

**PENGEMBANGAN SISTEM INFORMASI IZIN KERJA DAN
PRAKTIK TENAGA KESEHATAN (SINKES)
DINAS KESEHATAN KOTA XYZ**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Rosalia Indah

NIM: 145150401111083



**PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI PENGAJUAN IZIN KERJA DAN PRAKTIK
TENAGA KESEHATAN (SINKES) DINAS KESEHATAN KOTA XYZ

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Rosalia Indah

NIM: 145150401111083

Skripsi ini telah diuji dan dinyatakan lulus pada
25 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

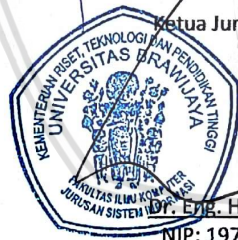
Dosen Pembimbing II

Yusi Tyroni Mursityo, S.Kom., M.AB
NIP: 19800228 200604 1 001

M. Chandra Saputra, S.Kom, M.T, M.Eng
NIK: 20160986010610010

Mengetahui

Ketua Jurusan Sistem Informasi



Eng. Herman Tolle, S.T., M.T
NIP: 19740823 200012 1 001

Tanggal Ujian dan Profil Penguji

Hari/Tanggal : RABU / 25 JULI 2018
Tempat : FILKOM UB / E2.2
Waktu : 14:00 – 15:15
Nama Penguji 1 : Fajar Pradana, S.ST., M.Eng
NIP : 19871121 2015041 004
Nama Penguji 2 : Aryo Pinandito, S.T., M.MT.
NIP : 19830519 2014041 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 25 Juli 2018



Rosalia Indah

NIM: 145150401111083

Curriculum Vitae Penulis**PERSONAL INFORMATION**

Nama : Rosalia Indah
 NIM : 145150401111083
 Prodi : Sistem Informasi
 Alamat di Malang : Jl. Mayjend Pandjaitan Gang 18 C
 No. 8 Malang
 Alamat Asal : Kranggan Gang 2 No. 21 RT 2
 RW 1 Mojokerto
 No. telepon : 085733289408
 Email : rosaliaindah389@gmail.com
 Jenis Kelamin : Perempuan
 Tempat, Tanggal lahir : Mojokerto, 8 Februari 1996

EDUCATION AND TRAINING**Formal**

2002 – 2008 SD Negeri Kranggan 5 Kota Mojokerto
 2008 – 2011 SMP Negeri 1 Kota Mojokerto
 2011 – 2014 SMA Negeri 3 Kota Mojokerto
 2014 – 2018 S1 Sistem Informasi Fakultas Ilmu Komputer
 Universitas Brawijaya

WORK EXPERIENCE

2014 – 2015 - Staff Pengembangan Sumber Daya Manusia EKSEKUTIF
 MAHASISWA SISTEM INFORMASI
 2015 – 2016 - Asisten Praktikum Sistem Operasi Laboratorium Jaringan Komputer
 2015
 - Staff divisi Danus PTIIK Award 2015
 - Asisten Praktikum Pemrograman Pemrograman Lanjut
 Laboratorium Komputasi Dasar 2016
 2016 – 2017 - Staff divisi Acara I-Fest 3.0
 - Staff divisi Danus Seminar Nasional Gamosphere
 - Asisten Praktikum Basis Data Laboratorium Sistem Informasi 2016-
 2017
 - Bendahara 3 Eksekutif Mahasiswa Sistem Informasi
 2017 – 2018 - Bendahara 1 Eksekutif Mahasiswa Sistem Informasi

PERSONAL SKILLS

Communication skills Indonesia, English

Computer skills **Bahasa Pemrograman**
 Java, PHP, HTML, CSS

MS Office Operation

Ms. Word, Ms. Excel, Ms. Powerpoint

Database

MySQL, DB2

KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat yang diberikan sehingga penulis dapat menyelesaikan skripsi berjudul “Pengembangan Sistem Informasi Izin Kerja Dan Praktik Tenaga Kesehatan (SINKES) Dinas Kesehatan Kota XYZ”

Penelitian ini tidak dapat penulis selesaikan tanpa didukung oleh ilmu yang didapatkan di kelas perkuliahan dan di saat yang sama juga mengajarkan banyak ilmu lain yang belum penulis dapatkan di kelas perkuliahan. Selain itu, penelitian ini juga memberikan penulis pengalaman untuk dapat berkomunikasi dengan banyak orang, menggali permasalahan yang ada, serta menyamakan persepsi dengan kedua dosen pembimbing. Proses penelitian yang dilakukan juga mengajarkan penulis untuk menjadi pribadi yang sabar dan senantiasa berbagi hal-hal bermanfaat yang berkaitan dengan kebutuhan pembelajaran. Penelitian ini tidak dapat diselesaikan tanpa dukungan dan bantuan dari orang-orang terdekat, sehingga saya ingin menyampaikan rasa terima kasih kepada :

1. Bapak Yusi Tyroni Mursityo, S.Kom., M.AB, selaku pembimbing satu yang telah memberikan waktu, tenaga dan pemikiran-pemikiran yang solutif kepada penulis selama masa pengerjaan penelitian, serta menjadi pembimbing yang sabar dan berdedikasi tinggi dalam memberikan arahan.
2. Bapak Mochamad Chandra Saputra, S.Kom, M.T, M.Eng, selaku pembimbing dua yang telah memberikan waktu, tenaga dan pemikiran-pemikiran yang solutif kepada penulis selama masa pengerjaan penelitian, serta menjadi pembimbing yang dapat berkomunikasi dengan baik dalam menyampaikan masukan.
3. Bapak Suprpto, S.T, M.T selaku Ketua Program Studi Sistem Informasi Universitas Brawijaya Malang.
4. Bapak Herman Tolle, Dr. Eng., S.T, M.T selaku Ketua Jurusan Sistem Informasi Universitas Brawijaya Malang.
5. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Almh. Ibu yang telah menjadi inspirasi utama penulis untuk menjadi wanita yang tangguh, mandiri dan bermanfaat bagi sekitar telah sangat berperan besar terhadap usaha penulis dalam menjalani kehidupan perantauan. Ayah dan kedua kakak di Mojokerto yang senantiasa memberikan dukungan berupa doa, moril dan materiil sehingga menjadi penyemangat penulis dalam menjalani perkuliahan dan kehidupan mandiri di Malang.
7. Pimpinan dan pegawai CV. Sarana Utama Solusindo, selaku narasumber yang telah memberikan informasi terkait permasalahan di Dinas Kesehatan Kota XYZ dan dokumen-dokumen pendukung penelitian.

8. Danniara Reza Firdausy, selaku sahabat penulis yang telah menjadi kawan yang sangat baik dalam mengembangkan kemampuan diri selama menjalani kehidupan perkuliahan, serta menjadi rekan yang dapat diandalkan untuk berdiskusi terkait berbagai hal. Yhouga Beta Evariantio, selaku kakak tingkat yang dengan sabar membagikan ilmunya untuk berdiskusi dalam menyelesaikan pengerjaan skripsi. Firnanda Iftitah Dewi Antono dan Edwin Nurwansyah yang telah menjadi teman untuk berdiskusi dan berbagi ilmu selama menjalani perkuliahan hingga pengerjaan skripsi.
9. Dhimas, Tedja, Aziz, Savira, Windha, Ozzy, Shella, Febiko, Caca, Higam selaku teman-teman yang senantiasa memberikan dukungan dan energi positif kepada penulis agar selalu menjadi pribadi yang optimis.
10. Teman-teman Eksekutif Mahasiswa Sistem Informasi dan Badan Perwakilan Mahasiswa Sistem Informasi periode 2015/2016, 2016/2017 dan 2017 selaku rekan kerja dalam kepengurusan organisasi yang telah memberikan kesempatan penulis untuk mengembangkan diri dalam kemampuan bersosialisasi dan berdiskusi.
11. Teman-teman Asisten Sistem Operasi periode 2015, Asisten Pemrograman Lanjut periode 2016 dan Asisten Praktikum Basis Data periode 2016-2015 serta 2017-2018 yang telah menjadi media penulis dalam belajar dan berdiskusi untuk kepentingan berbagi ilmu yang didapat selama masa perkuliahan berlangsung.
12. Teman-teman Sistem Informasi Universitas Brawijaya angkatan 2014 yang tidak dapat penulis sebutkan namanya satu per satu, yang telah menjadi teman yang baik selama menjalani perkuliahan di FILKOM UB.

Semoga Allah SWT melimpahkan rahmat berlipat ganda kepada seluruh pihak yang telah berkontribusi pada proses pengerjaan penelitian ini. Tidak lupa penulis sampaikan bahwa penelitian yang telah dilakukan memiliki banyak kekurangan, diskusi mengenai penelitian lanjutan, saran dan kritik yang membangun merupakan hal-hal yang penulis harapkan dapat disampaikan kepada penulis.

Malang, 9 Juli 2018



Penulis

rosaliindah389@gmail.com

ABSTRAK

Dinas Kesehatan (Dinkes) Kota XYZ yang dipimpin oleh seorang Kepala Dinas memiliki tanggungjawab terhadap pelaksanaan keperluan bidang kesehatan dalam lingkungan administrasi Pemerintah Kota XYZ. Sehubungan dengan tugasnya, Dinkes memiliki pelayanan pemberian surat izin kerja/praktik kepada Tenaga Kesehatan. Dalam pelaksanaan layanan tersebut, Tenaga Kesehatan perlu mendatangi Kantor Dinkes secara langsung lebih dari sekali jika berkas yang dibawa kurang atau salah, karena permohonan tidak akan diproses jika belum memenuhi ketentuan. Selain itu, dari sisi pihak Dinkes yang menyebabkan penanganan terhambat adalah berkas cetak yang dilampirkan oleh pemohon harus diperiksa secara langsung oleh Petugas Proses, Kepala Seksi Kefarmasian, Kepala Seksi SDM dan Kepala Bidang, sehingga proses penanganan akan tertunda jika yang bersangkutan tidak berada di Kantor Dinkes. Berdasarkan permasalahan tersebut, maka dikembangkan sebuah sistem informasi pengajuan izin kerja dan praktik yang dapat digunakan Tenaga Kesehatan dan Dinkes Kota XYZ untuk mengajukan permohonan dan menangani permohonan yang telah diajukan. Sistem informasi ini juga memungkinkan Tenaga Kesehatan mengetahui tahapan penanganan yang sedang dilakukan oleh petugas Dinkes, sehingga informasi terkait pengurusan izin kerja dapat lebih transparan. Pada penelitian ini, *V-Model* dipilih sebagai metode pengembangan perangkat lunak yang dijadikan acuan selama melakukan pengembangan karena kebutuhan sistem telah terdefinisi secara jelas dan tidak ambigu. Selanjutnya, hasil implementasi fungsi-fungsi diuji menggunakan pengujian *responsiveness*, pengujian *compatibility* dan pengujian penerimaan pengguna (UAT). Dari hasil pengujian *responsiveness*, diketahui bahwa sistem dapat ditampilkan dengan baik pada tiga jenis ukuran resolusi layar. Pada pengujian *compatibility* diketahui bahwa terdapat critical issue yang muncul akibat tidak diaktifkannya fitur ssl atau https. Kemudian dari hasil UAT diketahui bahwa sistem dapat diterima dengan baik oleh pengguna.

Kata kunci: Sistem informasi, surat izin, *V-Model*, *compatibility*, UAT

ABSTRACT

Department of health XYZ City led by a head of Department has a responsibility towards the implementation of the purposes of environmental health in the field of the administration of the City Government of XYZ. In connection with its work, the Health Office has a service of granting work permit or practice to health care personnel. In the performance of those services, health workers need to come to the Office Health Office directly more than once if it brought less or wrong because the application will not be processed if it does not meet the conditions. In addition, the health office party which led to handling the print file is inhibited attached by the applicant should be checked directly by the attendant process, Section Kefarmasian, Section Head of human resources and the head of the field, so that the process of handling will be delayed if the question was not in the Office Health Office. Based on these problems, then developed a system of filing work permit information and practices that can be used with healthcare personnel and Health Office city of XYZ to apply for and handle the petition has been filed. This information system also allows health workers to know the stages of the handling is being done by the officer's Health Office, so information related work permit arrangements can be more transparent. In this study, the V-Model was chosen as a method of software development made a reference during development because the system requirements have been clearly defined and unambiguous. Furthermore, the results of the implementation of the functions tested using responsiveness testing, compatibility testing, user acceptance testing and (UAT). From the results of testing the responsiveness, it is known that a system can be displayed properly on three types of screen resolution size. On compatibility testing, there is a critical issue that arises due to ssl or https feature is not enabled. Then from the UAT results note that system can be well received by users.

Keywords : Information system, permit, V-Model, compatibility, UAT

DAFTAR ISI

KATA PENGANTAR	vi
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI	x
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xviii
DAFTAR LAMPIRAN	xxi
BAB 1 PENDAHULUAN	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	3
1.3 Tujuan.....	3
1.4 Manfaat	4
1.5 Batasan masalah	4
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN.....	7
2.1 Tinjauan Pustaka	7
2.2 <i>Software Development Life Cycle (SDLC)</i>	11
2.2.1 <i>V-Model</i>	11
2.3 Pemrograman berorientasi obyek	14
2.4 Teknik pengumpulan data	14
2.5 <i>Level of requirement</i>	15
2.5.1 <i>Business perspective</i>	16
2.5.2 <i>User perspective</i>	18
2.6 <i>Bussines Processs Model Notation (BPMN)</i>	19
2.7 <i>Unified Modelling Languange (UML)</i>	23
2.7.1 <i>Use case diagram</i>	23
2.7.2 <i>Use case scenario</i>	25
2.7.3 <i>Activity diagram</i>	27
2.7.4 <i>Sequence diagram</i>	29
2.7.5 <i>Class diagram</i>	30
2.8 <i>Conceptual Data Model (CDM)</i>	31

2.9 <i>Physical Data Model (PDM)</i>	32
2.10 <i>Pola Model-View-Controller</i>	33
2.10.1 <i>Framework CodeIgniter</i>	33
2.11 <i>Pengujian</i>	34
2.11.1 <i>User Acceptance Testing (UAT)</i>	34
BAB 3 <i>METODOLOGI</i>	43
3.1 <i>Metodologi penelitian</i>	43
3.1.1 <i>Wawancara</i>	43
3.1.2 <i>Studi literatur</i>	44
3.1.3 <i>Analisis kebutuhan</i>	44
3.1.4 <i>Spesifikasi kebutuhan</i>	44
3.1.5 <i>Desain arsitektur sistem</i>	45
3.1.6 <i>Detil desain sistem</i>	45
3.1.7 <i>Implementasi</i>	45
3.1.8 <i>Pengujian sistem</i>	45
3.1.9 <i>Kesimpulan dan saran</i>	46
BAB 4 <i>ANALISIS KEBUTUHAN</i>	47
4.1 <i>Business perspective</i>	47
4.1.1 <i>Analisis proses bisnis as-is</i>	47
4.1.2 <i>Analisis permasalahan</i>	59
4.1.3 <i>Identifikasi proses bisnis to-be</i>	61
4.1.4 <i>Analisis perbaikan proses bisnis</i>	74
4.1.2 <i>Analisis kebutuhan pemangku kepentingan</i>	75
4.2 <i>User Perspective</i>	78
4.2.1 <i>Identifikasi Aktor</i>	78
4.2.2 <i>Identifikasi fitur</i>	79
4.3 <i>Spesifikasi kebutuhan pengembangan</i>	80
4.3.1 <i>Kebutuhan fungsional</i>	80
4.3.2 <i>Kebutuhan non-fungsional</i>	82
4.3.3 <i>Spesifikasi kebutuhan perangkat instalasi</i>	83
4.3.4 <i>Use case diagram</i>	83
4.3.5 <i>Use case scenario Tenaga Kesehatan</i>	85

4.3.6 Use case scenario Petugas Dinkes	93
4.4 Activity diagram	103
4.4.1 Activity diagram mengajukan permohonan	103
4.4.2 Activity diagram menyetujui pengajuan permohonan	104
4.4.3 Activity Diagram membuat jadwal kunjungan.....	105
4.4.4 Activity diagram menugaskan pembuatan surat tugas.....	106
BAB 5 PERANCANGAN SISTEM	108
5.1 Perancangan arsitektur sistem	108
5.2 Sequence diagram	108
5.2.1 Sequence diagram mengajukan permohonan	108
5.2.2 Sequence diagram menyetujui pengajuan permohonan	110
5.2.3 Sequence diagram membuat jadwal kunjungan.....	111
5.2.4 Sequence diagram menugaskan pembuatan surat tugas.....	112
5.3 Perancangan class diagram	113
5.3.1 Diagram kelas analisis	113
5.3.2 Diagram kelas perancangan	114
5.4 Perancangan basisdata	118
5.4.1 Perancangan <i>Conceptual Data Model</i> (CDM)	118
5.4.2 Perancangan <i>Physical Data Model</i> (PDM)	118
5.5 Perancangan antarmuka	119
5.5.1 Perancangan antarmuka mengajukan surat izin.....	119
5.5.2 Perancangan antarmuka menyetujui pengajuan permohonan	120
5.5.3 Perancangan antarmuka membuat jadwal kunjungan	121
5.5.4 Perancangan antarmuka menugaskan pembuatan surat tugas	122
5.6 Perancangan algoritme	123
5.6.1 Algoritme mengajukan permohonan	123
5.6.2 Algoritme menyetujui pengajuan permohonan.....	130
5.6.3 Algoritme membuat jadwal kunjungan	131
5.6.4 Algoritme menugaskan pembuatan surat tugas	131
BAB 6 IMPLEMENTASI	133
6.1 Implementasi program	133
6.1.1 Implementasi algoritme mengajukan permohonan.....	133

6.1.2 Implementasi algoritme menyetujui pengajuan permohonan .	146
6.1.3 Implementasi algoritme membuat jadwal kunjungan	150
6.1.4 Implementasi algoritme menugaskan pembuatan surat izin ...	151
6.2 Implementasi antarmuka	153
6.2.1 Implementasi halaman mengajukan permohonan	153
6.2.2 Implementasi halaman menyetujui pengajuan permohonan ..	154
6.2.3 Implementasi halaman membuat jadwal kunjungan	155
6.2.4 Implementasi halaman menugaskan pembuatan surat izin	156
BAB 7 PENGUJIAN	157
7.1 Perancangan pengujian sistem	157
7.1.1 <i>Responsiveness testing</i>	157
7.1.2 <i>Compatibility testing</i>	158
7.1.3 <i>User Acceptance Testing (UAT)</i>	159
7.6 Hasil pengujian	168
7.6.1 Hasil <i>responsiveness testing</i>	168
7.6.2 Hasil <i>compatibility testing</i>	173
7.6.3 Hasil <i>User Acceptance Testing (UAT)</i>	182
BAB 8 KESIMPULAN DAN SARAN	194
8.1 Kesimpulan	194
8.2 Saran	195
DAFTAR PUSTAKA	197
LAMPIRAN A HASIL WAWANCARA PEGAWAI CV. SARANA UTAMA SOLUSINDO	199
LAMPIRAN B PERNYATAAN PEMBERIAN DATA PENDUKUNG	201
LAMPIRAN C KUESIONER UAT TENAGA KESEHATAN	202
LAMPIRAN D KUESIONER UAT CV. SARANA UTAMA SOLUSINDO	206

DAFTAR TABEL

Tabel 2.1 Tinjauan pustaka	9
Tabel 2.2 Analisis Permasalahan	16
Tabel 2.3 Analisis Permasalahan yang Telah Mengalami Penyesuaian.....	16
Tabel 2.4 Analisis Tipe Pemangku Kepentingan.....	17
Tabel 2.5 Fitur Sistem	19
Tabel 2.6 Notasi pada BPMN.....	20
Tabel 2.7 Simbol <i>use case diagram</i>	24
Tabel 2.8 <i>Use case scenario</i>	25
Tabel 2.9 Tabel <i>use case scenario</i> yang telah mengalami penyesuaian.....	26
Tabel 2.10 Simbol-simbol <i>activity diagram</i>	28
Tabel 2.11 Simbol-simbol <i>sequence diagram</i>	29
Tabel 2.12 Simbol <i>Class Diagram</i>	30
Tabel 2.13 Simbol-simbol <i>Conceptual Data Model</i> (CDM)	32
Tabel 2.14 Kriteria Penerimaan UAT	37
Tabel 2.15 Tabel rancangan kasus uji validasi yang mengalami penyesuaian.....	38
Tabel 2.16 Tabel rancangan kasus uji penerimaan pengguna	38
Tabel 2.17 Tabel hasil pengujian penerimaan pengguna	39
Tabel 4.1 Analisis Permasalahan	59
Tabel 4.2 Analisis waktu proses yang dibutuhkan.....	60
Tabel 4.3 Perubahan aktivitas proses bisnis pengajuan permohonan.....	61
Tabel 4.4 Perubahan aktivitas proses bisnis penanganan permohonan izin praktik/kerja Tenaga Kesehatan.....	73
Tabel 4.5 Analisis perbedaan proses bisnis <i>as-is</i> dan <i>to-be</i> serta waktu yang dibutuhkan pada pengerjaan <i>task</i>	74
Tabel 4.6 Tipe pemangku kepentingan.....	75
Tabel 4.7 Kebutuhan pemangku kepentingan dan pengguna	76
Tabel 4.8 Tabel identifikasi aktor	78
Tabel 4.9 Identifikasi fitur	79
Tabel 4.10 Identifikasi kebutuhan fungsional	81
Tabel 4.11 Identifikasi kebutuhan non fungsional	83
Tabel 4.12 Identifikasi kebutuhan instalasi.....	83

Tabel 4.13 <i>Use case scenario</i> membuat akun	86
Tabel 4.14 <i>Use case scenario login</i>	87
Tabel 4.15 <i>Use case scenario</i> merubah biodata.....	88
Tabel 4.16 <i>Use case scenario</i> membuat berkas pendukung.....	89
Tabel 4.17 <i>Use case scenario</i> mengajukan permohonan	90
Tabel 4.18 <i>Use case scenario</i> melihat alur penanganan.....	91
Tabel 4.19 <i>Use case scenario</i> mencetak tanda bukti.....	92
Tabel 4.20 <i>Use case scenario logout</i>	93
Tabel 4.21 <i>Use case scenario login</i>	94
Tabel 4.22 <i>Use case scenario</i> menyetujui pengajuan permohonan	95
Tabel 4.23 <i>Use case scenario</i> menyetujui penanganan lebih lanjut	96
Tabel 4.24 <i>Use case scenario</i> membuat jadwal kunjungan	97
Tabel 4.25 <i>Use case scenario</i> melihat jadwal kunjungan	98
Tabel 4.26 <i>Use case scenario</i> membuat berita acara	99
Tabel 4.27 <i>Use case scenario</i> menugaskan pembuatan surat izin	100
Tabel 4.28 <i>Use case scenario</i> mencetak surat izin	101
Tabel 4.29 <i>Use case scenario logout</i>	102
Tabel 5.1 Penjelasan kelas-kelas pada kelas <i>logical</i>	115
Tabel 5.2 Penjelasan kelas-kelas pada kelas model	117
Tabel 5.3 Algoritme mengajukan permohonan	123
Tabel 5.4 Algoritme mengajukan permohonan untuk Apoteker	124
Tabel 5.5 Algoritme mengajukan permohonan untuk Bidan.....	125
Tabel 5.6 Algoritme mengajukan permohonan untuk Dokter	127
Tabel 5.7 Algoritme mengajukan permohonan untuk Perawat.....	128
Tabel 5.8 Algoritme mengajukan permohonan untuk Tenaga Teknis Kefarmasian	129
Tabel 5.9 Algoritme menyetujui pengajuan permohonan	131
Tabel 5.10 Algoritme membuat jadwal kunjungan pada kelas C_jadwal_kunjungan.....	131
Tabel 5.11 Algoritme menugaskan pembuatan surat izin	132
Tabel 6.1 Implementasi algoritme mengajukan permohonan	133
Tabel 6.2 Implementasi algoritme mengajukan permohonan Apoteker	134
Tabel 6.3 Implementasi algoritme mengajukan permohonan Bidan.....	135

Tabel 6.4 Implementasi algoritme mengajukan permohonan Dokter.....	138
Tabel 6.5 Implementasi algoritme mengajukan permohonan untuk Perawat...	139
Tabel 6.6 Implementasi algoritme mengajukan permohonan untuk Tenaga Teknis Kefarmasian.....	141
Tabel 6.7 Penjelasan implementasi algoritme mengajukan permohonan.....	143
Tabel 6.8 Penjelasan implementasi algoritme mengajukan permohonan Apoteker	143
Tabel 6.9 Penjelasan implementasi algoritme mengajukan permohonan Bidan.....	144
Tabel 6.10 Penjelasan implementasi algoritme mengajukan permohonan Dokter	144
Tabel 6.11 Penjelasan implementasi algoritme mengajukan permohonan Perawat	145
Tabel 6.12 Penjelasan implementasi algoritme mengajukan permohonan Tenaga Teknis Kefarmasian	145
Tabel 6.13 Implementasi algoritme menyetujui pengajuan permohonan.....	146
Tabel 6.14 Penjelasan implementasi algoritme menyetujui pengajuan permohonan.....	149
Tabel 6.15 Implementasi algoritme membuat jadwal kunjungan	150
Tabel 6.16 Penjelasan implementasi algoritme membuat jadwal kunjungan....	151
Tabel 6.17 Implementasi menugaskan pembuatan surat izin	151
Tabel 6.18 Penjelasan implementasi algoritme menugaskan pembuatan surat izin	152
Tabel 7.1 Rancangan kasus Uji <i>responsiveness testing</i>	157
Tabel 7.2 Rancangan kasus uji <i>compatibility testing</i>	158
Tabel 7.3 Rancangan kasus uji UAT validasi aktor Tenaga Kesehatan	161
Tabel 7.4 Rancangan kasus uji UAT validasi fungsional aktor Petugas Proses; Kepala Seksi Kefarmasian, Kepala Seksi SDM; UPT Puskesmas Wilayah, Seksi Kesehatan Keluarga dan Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas	164
Tabel 7.5 Rancangan kasus uji UAT pengajuan permohonan	166
Tabel 7.6 Rancangan kasus uji UAT penanganan permohonan	167
Tabel 7.7 Hasil <i>responsiveness testing</i>	170
Tabel 7.8 Hasil <i>compatibility testing</i>	174
Tabel 7.9 Hasil uji UAT validasi aktor Tenaga Kesehatan	183
Tabel 7.10 Hasil uji UAT validasi fungsional aktor Petugas Proses; Kepala Seksi Kefarmasian, Kepala Seksi SDM; UPT Puskesmas Wilayah, Seksi Kesehatan	

Keluarga dan Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas.....	186
Tabel 7.11 Hasil kasus uji UAT pengajuan permohonan	188
Tabel 7.12 Tabel hasil kalkulasi skala <i>Likert</i> UAT pengajuan permohonan	189
Tabel 7.13 Hasil kasus uji UAT penanganan permohonan.....	190
Tabel 7.14 Tabel hasil kalkulasi skala <i>Likert</i>	192

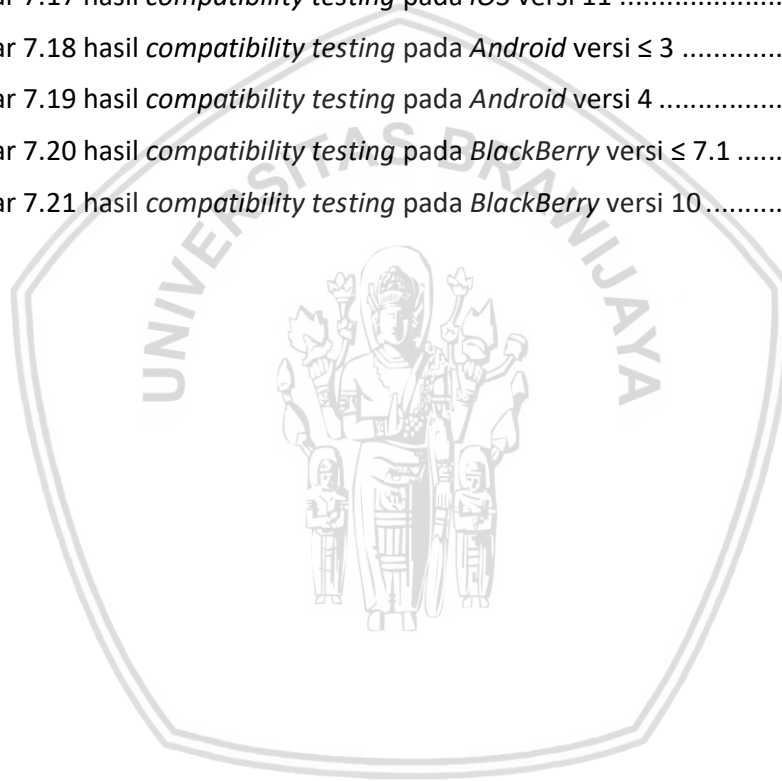


DAFTAR GAMBAR

Gambar 2.1 Tahapan dalam V-Model	12
Gambar 2.2 <i>Level of Requirement</i>	15
Gambar 2.3 <i>Use case diagram</i> untuk sistem music	24
Gambar 2.4 Contoh <i>Activity Diagram</i>	27
Gambar 2.5 Class diagram dari kelas Thoroughbred	31
Gambar 2.6 Contoh Diagram <i>Physical Data Mode</i>	33
Gambar 2.7 Alur kerja <i>Codeigniter</i>	34
Gambar 2.8 Contoh kasus uji validasi	38
Gambar 2.9 Tampilan pada website http://responsivedesignchecker.com/	41
Gambar 2.10 Contoh hasil <i>Compatibility Testing</i> menggunakan <i>Sortsite</i>	42
Gambar 3.1 Diagram alir metodologi penelitian	43
Gambar 4.1 Proses bisnis <i>as-is</i> pengajuan permohonan	48
Gambar 4.2 Proses bisnis <i>as-is</i> penanganan permohonan izin praktik Apoteker	50
Gambar 4.3 Proses bisnis <i>as-is</i> penanganan permohonan izin praktik Bidan	52
Gambar 4.4 Proses bisnis <i>as-is</i> penanganan permohonan izin praktik Perawat	54
Gambar 4.5 Proses bisnis <i>as-is</i> penanganan permohonan izin kerja Tenaga Teknis Kefarmasian	56
Gambar 4.6 Proses bisnis <i>as-is</i> penanganan permohonan izin praktik Dokter	58
Gambar 4.7 Pemodelan Proses Bisnis Pengajuan Permohonan <i>To-Be</i>	62
Gambar 4.8 Pemodelan proses bisnis pengajuan izin praktik Apoteker <i>to-be</i>	64
Gambar 4.9 Pemodelan proses bisnis pengajuan izin praktik Bidan <i>to-be</i>	66
Gambar 4.10 Proses bisnis pengajuan izin praktik Perawat <i>to-be</i>	68
Gambar 4.11 Proses bisnis pengajuan izin kerja Tenaga Teknis Kefarmasian <i>to-be</i>	70
Gambar 4.12 Proses bisnis pengajuan izin praktik Dokter <i>to-be</i>	72
Gambar 4.13 Contoh penomoran fitur dan deskripsinya	79
Gambar 4.14 Contoh penomoran kebutuhan fungsional maupun non fungsional dan deskripsi tiap penomoran	80
Gambar 4.15 <i>Use case diagram</i> Tenaga Kesehatan	84
Gambar 4.16 <i>Use case diagram</i> petugas Dinkes	85
Gambar 4.17 <i>Activity Diagram</i> Mengajukan Permohonan	104

Gambar 4.18 <i>Activity diagram</i> menyetujui berkas permohonan	105
Gambar 4.19 <i>Activity diagram</i> menambah jadwal kunjungan	106
Gambar 4.20 <i>Activity diagram</i> menugaskan pembuatan surat tugas.....	107
Gambar 5.1 <i>Architecture diagram</i> SINKES	108
Gambar 5.2 <i>Sequence diagram</i> mengajukan permohonan	109
Gambar 5.3 <i>Sequence diagram</i> menyetujui berkas permohonan	111
Gambar 5.4 <i>Sequence diagram</i> menambahkan jadwal kunjungan	111
Gambar 5.5 <i>Sequence diagram</i> menugaskan pembuatan surat tugas	112
Gambar 5.6 Diagram kelas analisis	113
Gambar 5.7 <i>class diagram</i> relasi antar kelas SINKES.....	114
Gambar 5.8 <i>Class diagram</i> controller SINKES	115
Gambar 5.9 <i>Class diagram</i> model website SINKES.....	117
Gambar 5.10 Perancangan <i>Conceptual Data Model</i> (CDM)	118
Gambar 5.11 Perancangan <i>Physical Data Model</i> (PDM)	119
Gambar 5.12 Perancangan antarmuka mengajukan surat izin.....	120
Gambar 5.13 Perancangan antarmuka menyetujui pengajuan permohonan....	121
Gambar 5.14 Perancangan antarmuka membuat jadwal kunjungan	122
Gambar 5.15 Perancangan antarmuka menugaskan pembuatan surat tugas ...	123
Gambar 6.1 Implementasi halaman mengajukan permohonan	153
Gambar 6.2 Implementasi halaman menyetujui pengajuan permohonan	154
Gambar 6.3 Implementasi halaman membuat jadwal kunjungan.....	155
Gambar 6.4 Implementasi halaman menugaskan pembuatan surat izin.....	156
Gambar 7.1 Pengkodean untuk menandai kasus uji pengujian UAT	159
Gambar 7.2 hasil <i>responsiveness testing</i> halaman pengajuan permohonan pada resolusi layar 1440*900	170
Gambar 7.3 hasil <i>responsiveness testing</i> halaman pengajuan permohonan pada resolusi layar 768*1024	171
Gambar 7.4 hasil <i>responsiveness testing</i> halaman pengajuan permohonan pada resolusi layar 414*736	172
Gambar 7.5 Hasil <i>compatibility testing</i>	173
Gambar 7.6 hasil <i>compatibility testing</i> pada IE versi 9.....	174
Gambar 7.7 hasil <i>compatibility testing</i> pada IE versi 10.....	174
Gambar 7.8 hasil <i>compatibility testing</i> pada IE versi 11.....	175

Gambar 7.9 hasil <i>compatibility testing</i> pada <i>Edge</i> versi 16	175
Gambar 7.10 hasil <i>compatibility testing</i> pada <i>Firefox</i> versi 60	176
Gambar 7.11 hasil <i>compatibility testing</i> pada <i>Safari</i> versi ≤ 10	176
Gambar 7.12 hasil <i>compatibility testing</i> pada <i>Safari</i> versi 11	177
Gambar 7.13 hasil <i>compatibility testing</i> pada <i>Opera</i> versi 51	177
Gambar 7.14 hasil <i>compatibility testing</i> pada <i>Chrome</i> versi 66	178
Gambar 7.15 hasil <i>compatibility testing</i> pada <i>iOS</i> versi ≤ 9	178
Gambar 7.16 hasil <i>compatibility testing</i> pada <i>iOS</i> versi 10	179
Gambar 7.17 hasil <i>compatibility testing</i> pada <i>iOS</i> versi 11	179
Gambar 7.18 hasil <i>compatibility testing</i> pada <i>Android</i> versi ≤ 3	180
Gambar 7.19 hasil <i>compatibility testing</i> pada <i>Android</i> versi 4	180
Gambar 7.20 hasil <i>compatibility testing</i> pada <i>BlackBerry</i> versi ≤ 7.1	181
Gambar 7.21 hasil <i>compatibility testing</i> pada <i>BlackBerry</i> versi 10	181



DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA PEGAWAI CV. SARANA UTAMA SOLUSINDO	199
A.1 Wawancara Pegawai CV. SARANA Utama Solusindo	199
LAMPIRAN B PERNYATAAN PEMBERIAN DATA PENDUKUNG	201
B.1 Surat pernyataan pemberian data pendukung oleh CV. Sarana Utama Solusindo	201
LAMPIRAN C KUESIONER UAT TENAGA KESEHATAN	202
C.1 Responden Tenaga Kesehatan 1.....	202
C.2 Responden Tenaga Kesehatan 2.....	203
C.3 Responden Tenaga Kesehatan 3.....	204
C.4 Responden Tenaga Kesehatan 4.....	205
LAMPIRAN D KUESIONER UAT CV. SARANA UTAMA SOLUSINDO	206
D.1 Responden pegawai CV. Sarana Utama Solusindo 1	206
D.2 Responden pegawai CV. Sarana Utama Solusindo 2	207
D.3 Responden pegawai CV. Sarana Utama Solusindo 3	209

BAB 1 PENDAHULUAN

1.1 Latar belakang

Dinas Kesehatan (Dinkes) merupakan unsur pelaksana bidang kesehatan dalam Pemerintahan Kota atau Kabupaten yang dipimpin oleh seorang Kepala Dinas yang bertanggungjawab kepada Sekretaris Daerah. Sebagai pelaksana urusan rumah tangga bidang kesehatan dalam organisasi pemerintahan, maka Dinas Kesehatan memiliki ikatan yang erat dengan para tenaga kesehatan. Dalam UU No. 39 Tahun 2009 tentang Kesehatan Bab 1 Pasal 1, dijelaskan bahwa *“Tenaga kesehatan adalah setiap orang yang mengabdikan diri dalam bidang kesehatan serta memiliki pengetahuan dan/atau keterampilan melalui pendidikan di bidang kesehatan yang untuk jenis tertentu memerlukan kewenangan untuk melakukan upaya kesehatan”*. Salah satu tugas Dinas Kesehatan adalah menyediakan pelayanan kepada tenaga kesehatan untuk mendapatkan surat izin kerja dan praktik sebagai syarat untuk melaksanakan pengabdian sesuai bidang ilmu yang dimiliki.

Dalam proses pengajuan izin kerja dan praktik di Dinas Kesehatan Kota XYZ, tenaga kesehatan diwajibkan untuk datang langsung ke Kantor Dinas Kesehatan dengan membawa berkas persyaratan yang dibutuhkan dan mengikuti prosedur yang ada. Berdasarkan SOP (*Standart Operational Procedure*) yang berlaku, permohonan tidak akan diproses oleh petugas jika berkas yang dibawa tidak lengkap sesuai persyaratan, sehingga hal ini menyebabkan pelayanan terhadap permohonan yang diajukan tertunda dan waktu yang dibutuhkan oleh tenaga kesehatan untuk mendapatkan izin menjadi lebih lama. Selain permasalahan tertundanya pelayanan yang disebabkan oleh kesalahan tenaga kesehatan sendiri, menurut informasi yang diberikan oleh pegawai CV. Sarana Utama Solusindo yang mendapat informasi langsung dari Petugas Proses (yang menangani permohonan izin kerja dan praktik) Dinkes Kota XYZ, lamanya pelayanan juga disebabkan oleh proses verifikasi berkas yang dilakukan bagian-bagian fungsional yang bersangkutan. Berkas yang diajukan oleh tenaga kesehatan wajib diperiksa oleh petugas beberapa bagian fungsional terkait sebelum akhirnya disetujui oleh Kepala Dinas untuk penerbitan surat izin. Berkas tersebut akan cepat diproses jika saat verifikasi data tidak ditemukan kesalahan dan petugas yang bertanggung jawab untuk memberikan persetujuan berada di Kantor Dinkes. Jika petugas tersebut tidak berada di kantor, maka proses pemeriksaan berkas akan tertunda.

Selanjutnya, terdapat permasalahan terkait pelaporan penanganan permohonan yang sedang ditangani kepada Kepala Dinas selaku pemimpin tertinggi organisasi dan pengarsipan berkas pemohon setelah proses permohonan izin selesai. Diketahui dari keterangan Petugas Proses, selama ini masih belum ada sarana yang digunakan Kepala Dinkes dalam mengawasi proses penanganan permohonan, sehingga Kepala Dinkes tidak bisa mengetahui sampai pada tahap apa permohonan di tangani. Terkait pengersipan berkas, ketika

tenaga kesehatan mendapatkan surat izin kerja/pembukaan praktik, berkas berupa lembaran kertas akan disimpan oleh pihak Dinkes di Gudang Arsip. Setiap periode tertentu, Petugas Proses akan melakukan rekap terhadap data permohonan izin yang masuk dengan mencari satu persatu berkas di Gudang Arsip. Selain proses pencarian yang membutuhkan waktu, berkas yang tersimpan rawan terselip dan hilang. Jika ada berkas yang tidak ditemukan, maka akan menghambat petugas dalam melakukan rekap karena tidak ada salinan lain dari berkas tersebut.

Berdasarkan masalah di atas, dibutuhkan sebuah aplikasi sistem informasi yang dapat digunakan sebagai jembatan antara pihak Dinas Kesehatan Kota XYZ dan para tenaga kesehatan dalam proses pengajuan izin kerja dan praktik. Sistem informasi tersebut diharapkan dapat membantu pegawai Dinkes yang bertugas untuk melakukan manajemen permohonan, melakukan validasi berkas, memberikan persetujuan untuk ke tahapan berikutnya, serta memberikan informasi kepada tenaga kesehatan bahwa permohonan izin yang diajukan telah disetujui oleh Kepala Dinas yang berwenang. Selain itu, sistem juga diharapkan mampu menjadi media bagi tenaga kesehatan dalam mengirimkan berkas pengajuan, agar mereka tidak perlu pulang dan berulang datang ke Kantor Dinkes jika ternyata ada berkas yang tertinggal atau kurang. Sistem yang dapat menampung berkas tenaga kesehatan juga akan berguna untuk pengarsipan data-data terkait tenaga kesehatan yang telah mengajukan dan mendapat surat izin kerja dan praktik.

Farhan Ishami (Ishami, et al., 2017) pada penelitiannya yang mengangkat topik tentang pengembangan sistem informasi *self-service* pemesanan kamar hotel, penelitian ini masuk dalam kategori *Customer Relationship Management* (CRM) operasional berjenis *Service Automation* karena sistem informasi yang dikembangkan membantu pelanggan turut aktif dalam proses pelayanan dengan tujuan untuk memudahkan dan mempercepat layanan diberikan. Dalam hal ini, tenaga kesehatan dianggap sebagai pelanggan dari Dinas Kesehatan karena tenaga kesehatan membutuhkan dan memanfaatkan pelayanan yang disediakan oleh Dinkes. Dari penelitian sebelumnya tersebut ditemukan beberapa kesamaan dengan masalah yang diangkat oleh peneliti untuk mengembangkan Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan, yaitu pengembangan sistem informasi yang membantu pelanggan dalam melayani dirinya sendiri untuk mempercepat dan mengoptimalkan pelayanan yang ditawarkan oleh penyedia jasa, serta membantu pihak penyedia jasa dalam melakukan manajemen pemesanan.

Untuk membangun aplikasi perangkat lunak, terdapat tahapan-tahapan aktivitas yang digunakan sebagai pedoman oleh pengembang agar aktivitas yang dilakukan berjalan secara sistematis. Tahapan aktivitas tersebut dikenal dengan sebutan *Software Development Life Cycle* (SDLC). SDLC sendiri memiliki banyak jenis model pengembangan, namun secara umum aktivitas yang ada didalam SDLC adalah analisis kebutuhan, pemodelan sistem, implementasi sistem dan pengujian perangkat lunak (Sommerville, 2009).

Tahap analisis kebutuhan menjadi hal dasar dalam melakukan pengembangan perangkat lunak untuk menghindari pembangunan perangkat lunak yang tidak sesuai dengan kebutuhan konsumen. Pada tahap analisis kebutuhan, pengembang perlu mengidentifikasi dan mendokumentasikan kebutuhan konsumen yang diharapkan ada pada sistem dalam level bahasa yang dimudah dipahami oleh orang yang tidak paham akan pengembangan perangkat lunak untuk memudahkan proses validasi (Petersen, et al., 2009). Setelah proses validasi dari hasil analisis kebutuhan dan kebutuhan sudah didefinisikan dengan benar, maka tahapan selanjutnya yang dilalui adalah pemodelan sistem. Pada tahap pemodelan sistem, dilakukan proses desain sistem berdasarkan kebutuhan yang telah didefinisikan pada tahap sebelumnya. Pada penelitian ini, pemodelan dilakukan dengan pendekatan *Object-Oriented Programming* (OOP) yang memungkinkan untuk menggambarkan sistem dari sudut pandang obyek yang terlibat dalam sistem. Karena pendekatan yang digunakan adalah OOP, maka pemodelan dibuat dengan standarisasi *Unified Modeling Language* (UML).

Setelah pemodelan sistem dibuat, maka selanjutnya adalah tahap implementasi. Pada tahap ini, terjadi aktivitas menerjemahkan pemodelan (perancangan) sistem kedalam baris kode pemrograman hingga perangkat lunak dapat berfungsi. Untuk memastikan bahwa perangkat lunak yang telah dibuat sesuai dengan pemodelan dan kebutuhan dari konsumen, maka tahap berikutnya yang dilakukan adalah pengujian sistem.

Maka dengan latar belakang yang dijabarkan, penulis tertarik untuk melakukan penelitian dengan judul “Pengembangan Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan (SINKES) Dinas Kesehatan Kota XYZ”. Diharapkan hasil penelitian ini bisa digunakan sebagai acuan dalam mengembangkan sistem informasi yang mampu membantu penanganan permohonan surat ijin kerja dan praktik oleh tenaga kesehatan.

1.2 Rumusan masalah

Berdasarkan permasalahan yang telah dipaparkan pada latar belakang, rumusan masalah untuk penelitian ini adalah sebagai berikut:

1. Bagaimana hasil analisis kebutuhan Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan (SINKES)?
2. Bagaimana hasil perancangan Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan (SINKES)?
3. Bagaimana hasil implementasi Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan (SINKES)?
4. Bagaimana hasil pengujian Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan (SINKES)?

1.3 Tujuan

Tujuan yang ingin dicapai melalui penelitian ini adalah menyelesaikan permasalahan Dinas Kesehatan Kota XYZ berupa lamanya pelayanan terhadap

pemberian izin kerja dan praktik tenaga kesehatan, serta manajemen penyimpanan data terkait perizinan yang masih kurang baik dengan pengembangan sistem informasi dengan menggunakan *Software Development Life Cycle* berupa *V-Model*.

1.4 Manfaat

Manfaat yang ingin diharapkan dari adanya penelitian ini adalah sebagai berikut.

1. Dapat membantu dalam meningkatkan pelayanan terhadap masyarakat, khususnya tenaga kesehatan.
2. Dapat menambah nilai keunggulan organisasi.
3. Dapat menjadi acuan organisasi dalam mengembangkan sistem informasi manajemen perizinan tenaga kesehatan sebagai bentuk pendukung proses bisnis.

1.5 Batasan masalah

Batasan yang diberikan untuk penelitian ini adalah:

1. Sistem yang dirancang merupakan sistem informasi yang mampu menampung permohonan pengajuan izin kerja dan praktik oleh tenaga kesehatan dan membantu pihak Dinkes Kota XYZ dalam menangani permohonan yang diajukan.
2. Sistem dikembangkan menggunakan *SDLC V-Model*. Tahapan pada *SDLC V-Model* yang akan dilakukan adalah *Requirements Analysis*, *Specifications*, *Architectural Design*, *Detail Design*, *Coding* dan *Acceptance Testing*. Maka dari itu tahapan *Unit Testing*, *Integration Testing* dan *System Testing* tidak akan dibahas lebih lanjut dalam penelitian ini.
3. Implementasi sistem menggunakan bahasa pemrograman PHP, pola perancangan MVC (*Model-View-Controller*) dan kerangka kerja *CodeIgniter*.
4. Informasi pendukung terkait pengembangan sistem didapatkan secara tidak langsung dari Dinkes Kota XYZ melalui CV. Sarana Utama Solusindo.

1.6 Sistematika pembahasan

Sistematika pembahasan penelitian ini dibagi menjadi 8 bab dengan pembagian materi sebagai berikut:

1. Bab 1 Pendahuluan

Pada Bab 1 berikut dipaparkan mengenai latar belakang penelitian, rumusan masalah yang disusun, tujuan, manfaat, Batasan masalah serta sistematika pembahasan yang akan diselesaikan oleh penelitian ini.

2. Bab 2 Landasan Kepustakaan

Berlanjut pada Bab 2, pada bagian ini berisikan teori-teori yang dijadikan dasar dalam melakukan penelitian yang akan dilakukan. Teori-teori tersebut didapatkan dari kajian pustaka dari penelitian sejenis yang pernah dilakukan, buku, jurnal penelitian, artikel dan *website* milik organisasi yang memiliki otoritas dalam bidang ilmu yang berkaitan. Teori yang diperlukan diantaranya tentang penelitian sejenis yang pernah dilakukan sebelumnya, *Software Development Life Cycle (SDLC) V-Model*, pemodelan proses bisnis menggunakan BPMN, pemodelan kebutuhan dan perancangan sitem menggunakan UML, pola perancangan MVC, kerangka kerja *CodeIgniter*, metode pengujian sistem yang akan digunakan dalam penelitian ini dan teori lain yang terkait.

3. Bab 3 Metodologi Penelitian

Setelah menjabarkan teori-teori yang digunakan sebagai landasan kepustakaan pada Bab 2, maka Bab 3 membahas langkah-langkah aktivitas yang akan dilakukan dalam penelitian ini untuk menyelesaikan permasalahan yang telah disampaikan pada Bab 1.

4. Bab 4 Analisis Kebutuhan Sistem

Bab 4 membahas tentang analisis kebutuhan dari sistem yang akan dikembangkan. Adapun aktivitas yang akan dilakukan untuk menganalisis kebutuhan sistem adalah analisis proses bisnis yang sedang berjalan, permasalahan yang dihadapi organisasi, identifikasi perbaikan proses bisnis yang ditawarkan untuk menyelesaikan masalah, identifikasi kebutuhan *stakeholder* dan pengguna, identifikasi fitur untuk memenuhi kebutuhan *stakeholder* dan pengguna, hingga akhirnya kebutuhan sistem dapat teridentifikasi. Kebutuhan sistem yang telah diidentifikasi selanjutnya digambarkan dengan *use case diagram* dan dijelaskan dengan menggunakan *use case scenario*.

5. Bab 5 Perancangan

Dari kebutuhan sistem yang telah dimodelkan dan dijelaskan pada Bab 4, kemudian akan dilakukan pemodelan perancangan menggunakan bahasa UML seperti *sequence diagram* dan *class diagram*. Pada bagian ini juga dibahas mengenai perancangan basis data serta perancangan antarmuka pengguna.

6. Bab 6 Implementasi

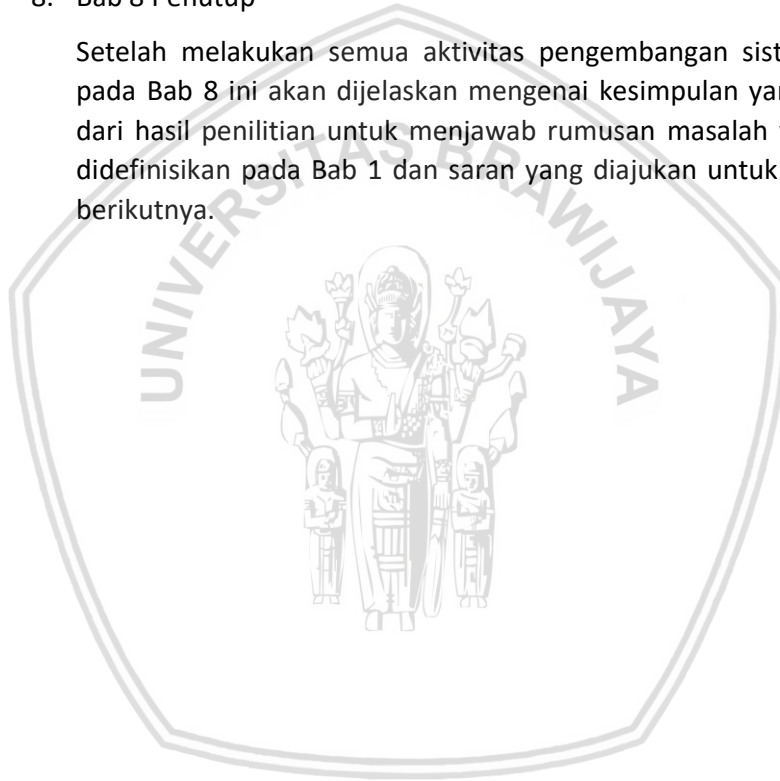
Bab 6 ini berisi pembahasan mengenai perancangan yang telah diimplementasikan menjadi kode program, penjelasan kode program dan implementasi dari perancangan antarmuka pengguna.

7. Bab 7 Pengujian Sistem

Setelah semua kebutuhan diimplementasikan menjadi kode program yang fungsionalnya telah berjalan, maka pada Bab 7 akan dijabarkan mengenai pengujian sistem untuk memastikan bahwa sistem yang dibuat telah sesuai dengan kebutuhan yang telah didefinisikan pada Bab 4. Adapun kegiatan pengujian terdiri dari perencanaan pengujian serta hasil yang didapatkan.

8. Bab 8 Penutup

Setelah melakukan semua aktivitas pengembangan sistem, maka pada Bab 8 ini akan dijelaskan mengenai kesimpulan yang didapat dari hasil penelitian untuk menjawab rumusan masalah yang telah didefinisikan pada Bab 1 dan saran yang diajukan untuk penelitian berikutnya.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Tinjauan Pustaka

Penelitian pertama yang dijadikan sebagai kajian pustaka pada adalah penelitian yang dilakukan oleh Farhan Ishami (Ishami, et al., 2017) dengan judul *"Pengembangan Sistem Self-Service Reservation Pada Everyday Smart Hotel Malang Menggunakan Customer Relationship Management (CRM) Operasional Modul Sales Force Automation dan Service Automation"*. Penelitian tersebut membahas tentang penerapan *self-service* untuk pemesanan layanan hotel seperti sewa kamar, *laundry* dan *ballroom*. Masalah yang diangkat adalah permasalahan yang terjadi pada proses bisnis pemesanan kamar, dimana petugas resepsionis harus terus menerus mengulang penawaran ke setiap pelanggan yang datang sehingga dianggap tidak akan efisien ketika terjadi antrian. Hasil penelitian berupa sistem informasi *self-service* digunakan untuk membantu pekerjaan petugas resepsionis dan mempercepat lamanya waktu pelayanan dengan mengikutsertakan peran aktif pelanggan untuk melakukan pemesanan kamar dan layanan lain melalui sistem tersebut sehingga antrian panjang dapat dihindari. Penelitian yang dilakukan (Ishami, et al., 2017) akan dijadikan sebagai referensi dalam menerapkan prinsip *Sales Force Automation dan Service Automation* pada pengembangan SINKES.

Kemudian ada penelitian yang dilakukan oleh (Nurjanah & Kurniadi, 2017) dengan judul *"Sistem Informasi Pengelolaan Izin Praktek Kerja Lapangan untuk Sekolah Menengah Kejuruan Secara Online di STT Garut"*. Permasalahan yang diangkat dalam penelitian berikut terkait dengan pengajuan izin praktek kerja lapangan oleh siswa Sekolah Menengah Kejuruan yang ingin melakukan kerja praktik di STT Garut guna memenuhi ketentuan akademik sekolah. Pada proses bisnis yang telah berjalan sebelumnya, siswa yang ingin mengajukan izin kerja praktik diwajibkan mendatangi kampus STT Garut untuk menyerahkan permohonan dan berkas yang berkaitan, sehingga hal ini akan menyulitkan para siswa yang memiliki tempat tinggal jauh dari lokasi kampus. Selain itu, diterima atau tidaknya permohonan praktik kerja yang diajukan tidak dapat diketahui siswa langsung setelah menyerahkan berkas, sehingga siswa perlu mendatangi kembali kampus STT Garut untuk melakukan konfirmasi kepada bagian administrasi. Maka dari itu, dilakukan penelitian untuk mengembangkan sebuah sistem informasi pengajuan izin yang dapat diakses secara *online* dari mana saja dengan harapan bahwa dapat membantu siswa dalam proses pendaftaran dan mengetahui keputusan kampus, serta dapat membantu staff administrasi dalam mengelola data pengajuan yang masuk. Pengembangan sistem informasi tersebut menggunakan metode pengembangan *Rational Unified Process* (RUP) dan pemodelan UML. Hasil identifikasi kebutuhan fungsional yang tertuang dalam diagram *usecase* pada jurnal penelitian *"Sistem Informasi Pengelolaan Izin Praktek Kerja Lapangan untuk Sekolah Menengah Kejuruan Secara Online di STT Garut"*, dijadikan sebagai pertimbangan oleh penelitian ini dalam menentukan kebutuhan fungsional yang perlu ada sebuah sistem informasi pengajuan izin.

Selanjutnya terdapat penelitian yang dilakukan oleh (Sharma & Singh, 2015) dengan judul "*Comparative Study of Various SDLC Models on Different Parameters*". Dalam pengembangan sebuah perangkat lunak, *Software Development Life Cycle* (SDLC) merupakan bagian penting yang dapat mempengaruhi kesuksesan sebuah proyek pengembangan perangkat lunak. Terdapat berbagai macam SDLC, diantaranya *Waterfall*, *Prototype*, *RAD (Rapid Application Development)*, *Incremental*, *Spiral* dan *V-Model*. Beragamnya model yang ada dalam pengembangan perangkat lunak membuat Sharma & Singh melakukan penelitian dengan menjelaskan definisi masing-masing model kemudian membandingkannya menggunakan beberapa parameter dengan tujuan untuk menunjukkan kelebihan maupun kekurangan dari setiap model. Hasil dari penelitian Sharma & Singh yang berupa penjelasan dan hasil perbandingan tiap model digunakan sebagai referensi pada penelitian ini untuk memilih SDLC yang tepat guna diimplementasikan dalam proses pengembangan SINKES.

Menurut (Sommerville, 2009) pada bukunya yang berjudul "*Software Engineering*" terdapat 4 aktivitas dasar yang harus ada dalam merekayasa perangkat lunak, meskipun ada berbagai *software processes* yang berbeda. Berikut adalah 4 aktivitas dasar tersebut.

1. *Software specification*
Mendefinisikan fungsionalitas perangkat lunak dan batasan pada penggunaannya.
2. *Software design and implementation*
Harus memproduksi perangkat lunak yang memenuhi spesifikasi.
3. *Software validation*
Perangkat lunak harus divalidasi untuk memastikan bahwa telah memenuhi keinginan dari konsumen.
4. *Software evolution*
Perangkat lunak harus bisa berevolusi untuk memenuhi perubahan pada kebutuhan pelanggan.

Tabel 2.1 Tinjauan pustaka

No.	Penelitian (Nama Jurnal, Judul Jurnal, Nama Peneliti)	Tujuan Penelitian	Metode Penelitian	Hasil Penelitian
1	<ul style="list-style-type: none"> - <u>Nama peneliti :</u> Ishami, Farhan; Rokhmawati, Retno Indah; Saputra, Mochammad Chandra - <u>Judul jurnal :</u> Pengembangan Sistem Self-Service Reservation pada Everyday Smart Hotel Malang Menggunakan <i>Customer Relationship Management</i> (CRM) Operasional Modul <i>Sales Force Automation</i> dan <i>Service Automation</i> - <u>Nama jurnal :</u> Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer - <u>Tahun terbit :</u> 2017 	<ul style="list-style-type: none"> - Membuat sistem informasi reservasi kamar hotel dan layanan lainnya dari <i>Everyday Smart Hotel</i> yang memungkinkan pelanggan untuk melakukan pemesanan secara mandiri (<i>self-service</i>). - Menerapkan prinsip-prinsip CRM operasional bagian <i>Service Automation</i> dan <i>Sales Forced automation</i> pada pengembangan sistem yang dilakukan. - Melakukan evaluasi dari hasil implementasi sistem untuk mengetahui perbedaan yang terjadi antar sebelum dan sesudah adanya sistem. 	<ul style="list-style-type: none"> - Prinsip-prinsip CRM pada bagian <i>Service Automation</i> dan <i>Sales Forced Automation</i>. - Sistem dikembangkan dengan pendekatan <i>object-oriented</i>, sehingga pemodelan kebutuhan dan perancangannya digambarkan dengan <i>Unified Modeling Language</i> (UML). - Pengujian sistem dilakukan dengan metode pengujian <i>white-box</i> berupa <i>basis path testing</i> dan <i>black-box</i> berupa <i>usability testing</i>. 	<ul style="list-style-type: none"> - Sistem pemesanan kamar dan layanan hotel yang dapat mempercepat proses pelayanan pihak hotel kepada pelanggan. - Sistem yang dihasilkan telah melalui pengujian dengan hasil yang baik, sehingga dapat disimpulkan bahwa sistem informasi yang dibangun telah sesuai dengan kebutuhan harapan pihak hotel dan pelanggan.
2	<ul style="list-style-type: none"> - <u>Nama peneliti :</u> Nurjanah, Aneu Suci; Kurniadi, Dede - <u>Judul jurnal :</u> Sistem Informasi Pengelolaan Izin Praktek Kerja Lapangan untuk 	<ul style="list-style-type: none"> - Mengembangkan sebuah sistem informasi yang bertujuan untuk menyelesaikan permasalahan pada proses bisnis pengajuan izin praktik kerja lapangan di Sekolah Tinggi Teknologi Garut. - Menerapkan SDLC RUP dengan menggunakan pendekatan berorientasi obyek. 	<ul style="list-style-type: none"> - Pengembangan perangkat lunak yang dilakukan menggunakan pendekatan berorientasi obyek. - SDLC <i>Rational Unified Process</i> (RUP) 	<ul style="list-style-type: none"> - Sistem Informasi Pengelolaan Izin Praktek Kerja Lapangan untuk Sekolah Menengah Kejuruan di STT Garut.

Tabel 2.1 Tinjauan pustaka

No.	Penelitian (Nama Jurnal, Judul Jurnal, Nama Peneliti)	Tujuan Penelitian	Metode Penelitian	Hasil Penelitian
	<p>Sekolah Menengah Kejuruan Secara Online di STT Garut</p> <ul style="list-style-type: none"> - <u>Nama jurnal</u> : Jurnal STT Gresik - <u>Tahun terbit</u> : 2017 		<ul style="list-style-type: none"> - Pengujian sistem dengan menggunakan metode <i>black-box testing</i>. 	<ul style="list-style-type: none"> - Dokumentasi analisis kebutuhan, perancangan, implementasi sistem dan hasil pengujian dengan metode <i>blackbox</i>.
3	<ul style="list-style-type: none"> - <u>Nama peneliti</u> : Sharma, Prateek; Singh, Dhananjaya - <u>Judul jurnal</u> : <i>Comparative Study of Various SDLC Models on Different Parameters</i> - <u>Nama jurnal</u> : <i>International Journal of Engineering Research</i> - <u>Tahun terbit</u> : 2015 	<ul style="list-style-type: none"> - Memberikan pengetahuan tentang perbandingan 7 SDLC yang berbeda yaitu : <i>Waterfall, Prototype, Rapid Application Development (RAD), Incremental, Spiral, Build and Fix, serta V-Shape model</i>. - Adapun pengetahuan yang ingin disampaikan adalah definisi, kelebihan dan kekurangan yang dimiliki masing-masing SDLC tersebut sebagai pertimbangan dalam menentukan SDLC yang tepat untuk diimplementasikan pada pengembangan yang akan dilakukan. 	<ul style="list-style-type: none"> - Melakukan perbandingan menggunakan parameter berupa kemudahan dalam mendefinisikan kebutuhan, keikutsertaan pengguna dalam setiap fase, analisis resiko, pengulangan fase, waktu implementasi, biaya, penanganan perubahan, kesederhanaan dan fleksibilitas. 	<ul style="list-style-type: none"> - Hasil perbandingan dituliskan dalam sebuah tabel lengkap dengan nama model dan parameter yang digunakan. - Peneliti menyimpulkan bahwa tim pengembang perangkat lunak dapat menggunakan SDLC yang mana saja menyesuaikan dengan kebutuhan dari konsumen.

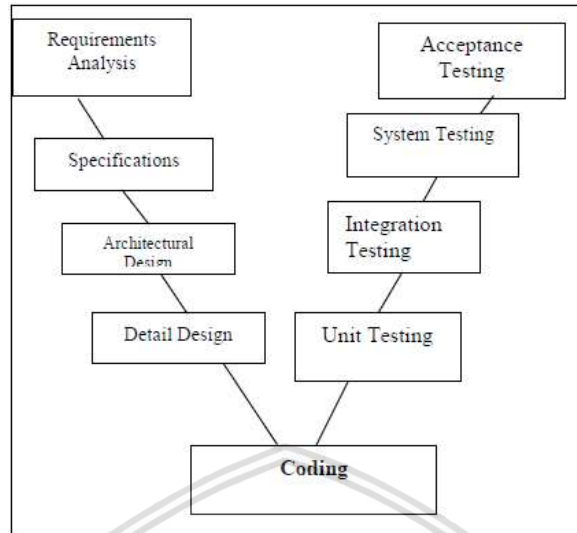
2.2 Software Development Life Cycle (SDLC)

Menurut Windi Eka (Eka, et al., 2013), *Software Development life cycle* (SDLC) adalah tahapan-tahapan yang dilakukan ketika mengembangkan sebuah perangkat lunak (*software*). Tahapan-tahapan aktivitas digunakan sebagai pedoman oleh pengembang agar aktivitas yang dilakukan berjalan secara sistematis. SDLC sendiri memiliki banyak jenis model pengembangan, namun secara umum aktivitas yang ada didalam SDLC adalah analisis kebutuhan, pemodelan sistem, implementasi sistem dan pengujian perangkat lunak (Sommerville, 2009). Didalam kerangka kerja SDLC terdapat beberapa model, salah satunya adalah *V-Model* yang dipilih sebagai SDLC dalam pengembangan SINKES.

2.2.1 V-Model

V-Model (*Validation & Verification model*) pertama kali diusulkan oleh Paul Rook pada tahun 1980. *V-Model* merupakan salah satu model SDLC yang dikembangkan dari model *Waterfall* dan memiliki bentuk menyerupai huruf "V". Tahapan-tahapan yang ada didalam *V-Model* hampir sama dengan *Waterfall*, hanya saja pada model ini tahapan pengujian dirinci untuk masing-masing tahapan (Eka, et al., 2013).

Meskipun *V-Model* adalah hasil pengembangan dari model *Waterfall*, namun model ini tidak dirancang dalam sumbu linier, melainkan tahapannya akan kembali keatas setelah tahan penulisan kode selesai. Menurut (Balaji, 2012), proses pengembangan dengan menggunakan *V-Model* bersifat seimbang dan bergantung pada verifikasi dari langkah sebelumnya sebelum melangkah maju, sehingga setiap tahapan perlu dilakukan pemeriksaan dan persetujuan sebelum berlanjut ke tahapan berikutnya. Balaji juga berpendapat bahwa di *V-Model*, *developer* dan *tester* dapat bekerja secara parallel. Dalam *V-Model*, bagian kiri dari bentuk "V" menunjukkan fase spesifikasi kebutuhan, bagian bawah (*tail*) menunjukkan fase pengembangan dan bagian kanan menunjukkan fase pengujian. Berikut adalah gambar yang menunjukkan tahapan pengembangan perangkat lunak di *V-Model* (Balaji, 2012).



Gambar 2.1 Tahapan dalam V-Model

Sumber : Mathur & Malik (2010)

Berikut penjelasan dari setiap fase pada *V-Model* menurut beberapa penelitian.

1. Analisis Kebutuhan (*Requirement Analysis*)

Pada tahap ini dilakukan analisis kebutuhan dengan mengumpulkan informasi dari pemangku kepentingan terkait tentang permasalahan yang dihadapi untuk mengetahui kebutuhan yang diharapkan dapat menjadi solusi permasalahan yang ada (Pressman, 2010).

2. Spesifikasi Kebutuhan (*Specifications*)

Spesifikasi kebutuhan adalah proses menuliskan kebutuhan pengguna dan sistem kedalam sebuah dokumen kebutuhan. Kebutuhan pengguna dan sistem seharusnya jelas, tidak ambigu, mudah dimengerti, lengkap dan konsisten. Kebutuhan pengguna untuk sebuah sistem mendeskripsikan kebutuhan fungsional dan non fungsional agar mudah dipahami oleh pengguna sistem yang tidak harus mengerti akan hal-hal teknis secara detil (Pressman, 2010).

3. Desain Arsitektur (*Architectural Design*)

Desain arsitektur merupakan proses kreatif ketika mendesain organisasi sistem yang akan memenuhi kebutuhan fungsional dan non fungsional sistem. Karena hal ini adalah sebuah proses kreatif, maka aktivitas di dalam proses bergantung pada tipe sistem yang akan dikembangkan, latar belakang dan pengalaman arsitek dan kebutuhan spesifik untuk sistem. Cara berpikir yang digunakan dalam mendesain arsitektur fokus terhadap serangkaian keputusan daripada aktivitas yang berkesinambungan. Selama proses mendesain, arsitek sistem harus menentukan jumlah

keputusan struktural yang akan mempengaruhi sistem dan proses pengembangannya.

4. Desain Detil (*Detail Design*)

Dari daftar kebutuhan yang telah dispesifikasi, selanjutnya dilakukan pemodelan yang lebih mendetail dan teknis. Berbeda dengan fase spesifikasi kebutuhan yang bahasanya mudah dipahami oleh pengguna yang awam akan hal teknis, pada desain detil bahasa yang digunakan akan sulit dimengerti oleh pengguna. Hasil dari detil desain ini nantinya akan digunakan sebagai dasar implementasi desain ke dalam bahasa pemrograman (Pressman, 2010).

5. Pengkodean (*Coding*)

Pengkodean merupakan aktivitas menerjemahkan desain teknis kedalam bahasa pemrograman. Fase ini harus dapat mewujudkan kebutuhan fungsional maupun non fungsional yang telah didefinisikan (Sommerville, 2009).

6. Pengujian Unit (*Unit Testing*)

Pengujian sistem yang berfokus pada masing-masing komponen secara individual, memastikan bahwa apakah sebuah fungsi berkerja dengan seharusnya sebagai sebuah unit. Menggunakan teknik pengujian *white-box*, pengujian dilakukan dengan mencoba jalur pada sebuah struktur modul untuk memastikan memenuhi *coverage* dan deteksi *error* maksimum (Mathur & Malik, 2010). Pada penelitian pengembangan SINKES, pengujian unit tidak dibahas lebih lanjut.

7. Pengujian Integrasi (*Integration Testing*)

Pengujian integrasi menangani perakitan dan integrasi komponen untuk membentuk sebuah perangkat lunak yang lengkap. Pengujian dilakukan dengan menggunakan teknik pengujian *black-box* untuk menyatukan permasalahan yang berhubungan menjadi permasalahan verifikasi dan konstruksi program (Mathur & Malik, 2010). Pada penelitian pengembangan SINKES, pengujian integrasi tidak dibahas lebih lanjut.

8. Pengujian Sistem (*System Testing*)

Pengujian dilakukan pada sebuah sistem yang terintegrasi dan lengkap untuk mengevaluasi apakah sistem telah memenuhi spesifikasi kebutuhan. Pengujian ini tidak memerlukan pengetahuan dalam desain kode atau logika pemrograman (Mathur & Malik, 2010). Pada penelitian pengembangan SINKES, pengujian integrasi tidak dibahas lebih lanjut.

9. Pengujian Penerimaan (*Acceptance Testing*)

Pengujian untuk memverifikasi apakah sebuah produk telah memenuhi kebutuhan yang telah dispesifikasikan oleh pelanggan. Seorang pelanggan

biasanya melakukan pengujian jenis ini pada produk yang telah dikembangkan (Mathur & Malik, 2010).

Dalam memilih SDLC yang akan digunakan pada pengembangan sebuah perangkat lunak, terdapat hal-hal yang perlu diperhatikan agar SDLC yang dipilih sesuai dengan karakteristik perangkat lunak yang akan dibangun. Menurut Sharma & Singh (2015), yang perlu diperhatikan sebelum mengimplementasikan *V-Model* adalah sebagai berikut.

1. Kebutuhan sudah terdefinisi dengan baik, dokumentasi jelas dan pasti.
2. Definisi dari produk tidak berubah-ubah.
3. Teknologi yang digunakan bersifat dinamis dan dimengerti dengan baik oleh pengembang.
4. Tidak ada kebutuhan yang ambigu dan tidak terdefinisi.

Pada Dari yang telah disebutkan diatas, maka *V-Model* dipilih sebagai SDLC dalam pengembangan SINKES karena kebutuhan telah jelas terdefinisi dan dokumen pendukung untuk pengembangan telah didapatkan.

2.3 Pemrograman berorientasi obyek

Pengembangan SINKES dilakukan dengan menggunakan pemrograman berorientasi obyek. Strategi utama pada pemrograman berorientasi obyek sesungguhnya adalah melihat sistem sebagai sekumpulan obyek yang berisi data dan fungsi operasi yang diberlakukan (Sukanto & Shalahuddin, 2016). Metode ini banyak dipilih karena memiliki keunggulan yaitu :

1. Obyek yang ditemukan dari masalah tertentu dapat dipakai ulang, sehingga dianggap mampu meningkatkan produktivitas.
2. Sistem yang dianalisis dan dirancang dengan baik dan benar akan berpengaruh pada berkurangnya resiko kesalahan saat kode diimplementasikan, sehingga dapat mempercepat proses pengembangan.
3. Model yang mudah dalam menerima perubahan, sehingga meningkatkan kemudahan dalam pemeliharaan.
4. Terdapat skema pewarisan atribut dan fungsi dari obyek yang bersifat general ke yang lebih spesifik. Dengan begitu, maka tidak perlu mendefinisikan atribut dan fungsi yang sama secara berulang.
5. Obyek dibuat berdasarkan kejadian pada dunia nyata, sehingga perangkat lunak yang dihasilkan dapat memenuhi kebutuhan pengguna.

2.4 Teknik pengumpulan data

Menurut Sugiyono (2016) bahwa dalam sebuah penelitian, perlu diadakan pengumpulan data dengan cara wawancara sebagai studi pendahuluan untuk menemukan permasalahan yang terjadi. Maka dari itu, untuk mengetahui informasi terkait prosedur pengajuan permohonan surat izin kerja/praktik oleh Tenaga Kesehatan dan penanganannya oleh pihak Dinkes serta permasalahan yang terjadi, maka dibutuhkan wawancara dengan narasumber dari CV. Sarana

Utama Solusindo selaku pihak yang dipercaya Dinkes Kota XYZ dalam pengembangan sistem informasi untuk mendukung kinerja Dinkes dalam menjalankan tugasnya. Teknik wawancara yang dipilih pada penelitian ini adalah wawancara tidak terstruktur. Wawancara tidak terstruktur merupakan jenis wawancara yang tidak menggunakan pedoman wawancara yang telah disusun secara lengkap dan sistematis, namun tetap berdasarkan pada garis-garis besar permasalahan yang ingin digali. Jenis wawancara ini sering digunakan untuk penelitian yang lebih mendalam tentang responden, karena peneliti tidak perlu terpaku pada pedoman wawancara tertentu, sehingga dapat fokus terhadap hal-hal yang diutarakan yang dapat dijadikan dasar pertanyaan berikutnya yang lebih terarah pada suatu tujuan (Sugiyono, 2016).

2.5 Level of requirement

Dalam melakukan pengembangan SINKES menggunakan *V-Model*, terdapat fase analisa kebutuhan dan spesifikasi seperti yang telah dijelaskan pada poin 2.2.1. Menurut IBM Corporation Rational Software (2011), dalam menganalisa kebutuhan perangkat lunak terdapat 2 jenis tingkatan sudut pandang, yaitu *Business Perspective* dan *User Perspective* seperti yang ditunjukkan pada Gambar 2.2. Aktivitas pada fase analisa kebutuhan dan spesifikasi akan dilakukan dengan mengacu pada *Level of Requirement*.



Gambar 2.2 Level of Requirement

Sumber : Diadaptasi dari IBM Corporation Rational Software (2011)

Pada gambar 2.2 dijelaskan tentang bagian-bagian yang harus didefinisikan dalam *requirements gathering*. Pada analisis *business perspective* bisa didapatkan masalah-masalah apa saja yang dihadapi oleh organisasi dengan mengetahui proses bisnis yang sedang berjalan (baik itu yang sudah ada maupun yang akan dibuat), dan apasaja kebutuhan dari pemangku kepentingan. Hasil analisis dari *business perspective* bisa dijawab dengan menggunakan analisa *user perspective* dengan memberikan solusi-solusi yang sesuai dengan kebutuhan dari pemangku kepentingan. Pada analisis ini dihasilkan fitur, rancangan kebutuhan fungsional

dan non fungsional dan rancangan sistem yang bisa memenuhi kebutuhan dari pemangku kepentingan.

2.5.1 Business perspective

Business perspective adalah sudut pandang yang melihat kebutuhan pengembangan aplikasi dari sisi fungsi dan proses bisnis untuk memenuhi tujuan bisnis yang ingin dicapai. Tujuannya adalah untuk memberikan pemahaman terhadap *stakeholder* bagaimana alur proses bisnis yang ada pada perusahaan tersebut yang bertujuan untuk mendapatkan gambaran secara jelas bagaimana proses bisnis yang sudah berjalan dan bagian mana dari proses bisnis tersebut yang akan ditingkatkan kinerjanya dengan menggunakan aplikasi tersebut sehingga mendapatkan kebutuhan yang mendukung tujuan bisnis yang ingin dicapai. *Business perspective* dapat disebut juga sebagai domain masalah.

1. Business process (as-is vs to-be)

Business process as-is menjelaskan secara keseluruhan proses bisnis yang berjalan saat ini di sebuah perusahaan atau organisasi dengan menggunakan pernyataan dan gambar. Tujuan dari analisis *Business process as-is* untuk memperjelas dan memahami seperti apa kondisi saat ini dari proses bisnis yang ada dan bagaimana alur bisnis tersebut berjalan pada perusahaan atau organisasi (Bridging the Gap, 2017). Proses analisis yang dilakukan untuk mengetahui permasalahan dan kelemahan yang ada pada proses bisnis tersebut. Permasalahan tersebut dipetakan dalam tabel analisis permasalahan, sehingga bisa diketahui bagaimana solusi dan perbaikan untuk proses bisnis tersebut. Tabel 2.2 merupakan cara dalam melakukan analisis permasalahan.

Tabel 2.2 Analisis Permasalahan

Masalah	[Mendeskripsikan masalah]
Mempengaruhi	[Pemangku kepentingan yang terpengaruh oleh masalah]
Dampak	[Dampak permasalahan]
Solusi	[Solusi beserta manfaat]

Sumber : Bittner & Spence (2002)

Untuk memperjelas masalah yang muncul dari proses bisnis yang berjalan saat ini di Dinkes, maka Tabel 2.2 mengalami penyesuaian dengan menambahkan pemangku kepentingan pemilik masalah seperti yang ditunjukkan pada Tabel 2.3 berikut.

Tabel 2.3 Analisis Permasalahan yang Telah Mengalami Penyesuaian

Masalah	Pemilik
[Mendeskripsikan masalah]	[Pemangku kepentingan pemilik masalah]
Mempengaruhi	
[Pemangku kepentingan yang terpengaruh oleh masalah]	
Dampak	
[Dampak permasalahan]	

Tabel 2.3 Analisis Permasalahan yang Telah Mengalami Penyesuaian

Solusi
[Solusi beserta manfaat]

Business process to-be menjelaskan bagaimana peningkatan kinerja proses bisnis kedepannya kepada *stakeholder* perusahaan atau organisasi. Usulan yang diajukan berdasarkan apa yang sudah dilakukan pada analisis *Business process as-is* untuk memenuhi seluruh permasalahan yang ada pada analisis *Business Process as-is*. Tujuan dari analisis *Business process to-be* adalah untuk memberikan penjelasan dan gambaran kepada *stakeholder* perusahaan atau organisasi bagaimana usulan-usulan tersebut disatukan dengan proses bisnis yang ada, apakah sudah memenuhi keinginan dari *stakeholder* yang terlibat dalam proses bisnis tersebut (Bridging the Gap, 2017).

2. Kebutuhan Pemangku Kepentingan

Menurut Sommerville dan Sawyer menjelaskan bahwa *stakeholder* atau pemangku kepentingan adalah seseorang yang mendapatkan manfaat baik itu secara langsung atau tidak langsung dari sistem yang dikembangkan (Somerville & Sawyer, 1997). Identifikasi kebutuhan dari pemangku kepentingan dilakukan untuk mengetahui bagaimana hubungan antara aktor dengan masalah-masalah yang ada sebelum diterapkannya aplikasi atau sistem. Dengan mengetahui siapa saja pemangku kepentingan yang akan menggunakan sistem tersebut, pengembang aplikasi bisa menentukan fungsi dan fitur apa saja yang ada dalam aplikasi dengan menyesuaikan karakteristik dari pemangku kepentingan tersebut, dimana pemangku kepentingan bertindak sebagai *user* atau pengguna dari sistem. Langkah-langkah yang dilakukan dalam analisis kebutuhan pemangku kepentingan adalah :

a. Tipe pemangku kepentingan

Tipe pemangku kepentingan merupakan kumpulan pemangku kepentingan yang memiliki karakteristik dan hubungan yang sama dengan sistem atau aplikasi yang dikembangkan. Pada tabel 2.4 merupakan cara dalam melakukan analisis tipe pemangku kepentingan.

Tabel 2.4 Analisis Tipe Pemangku Kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
[Tipe pemangku kepentingan]	[Menjelaskan tipe pemangku kepentingan]	[Menyebutkan pemangku kepentingan yang termasuk dalam tipe pemangku kepentingan]

Sumber: Diadaptasi dari Bittner & Spence (2002)

b. Kebutuhan pemangku kepentingan dan pengguna

Kebutuhan pemangku kepentingan menggambarkan masalah dalam proses bisnis atau pribadi dari pemangku kepentingan yang perlu untuk dilakukan analisis bagaimana solusi yang dibutuhkan untuk mendukung tujuan bisnis dan menyelesaikan masalah yang ada. Identifikasi kebutuhan

pemangku kepentingan akan memberikan pandangan mengenai masalah utama dan solusi yang dapat menyelesaikan masalah tersebut (Bittner & Spence, 2003).

2.5.2 User perspective

User perspective merupakan sudut pandang pengguna dimana didapatkan solusi bagi pengguna untuk mengatasi masalah yang ada di *business perspective*.

1. Fitur

Fitur merupakan suatu kemampuan sistem yang menyediakan nilai bagi pengguna serta memenuhi kebutuhan pemangku kepentingan dan pengguna. Suatu fitur dapat memiliki beberapa kebutuhan fungsional, karena secara sederhana fitur dapat diartikan sebagai penjelasan singkat dari sekumpulan perilaku sistem. Fitur tidak memiliki definisi yang jelas dan/atau level abstraksi yang konsisten untuk menggambarkan kemampuan sistem, maka dari itu penjelasan mengenai kemampuan sistem akan dijelaskan pada identifikasi kebutuhan sistem berupa kebutuhan fungsional dan non fungsional (Bittner & Spence, 2003). Fitur yang telah dibuat dilakukan prioritas dengan menggunakan metode MoSCow.

Salah satu teknik yang digunakan untuk prioritas kebutuhan adalah aturan MoSCow. Aturan MoSCow merupakan aturan yang membagi kebutuhan menjadi empat prioritas yaitu (Bittner & Spence, 2002) :

1. *Must Have* (Mo)
Must Have memiliki arti bahwa suatu fitur dalam proyek harus diselesaikan dan dapat bekerja dengan baik ketika proyek selesai. Fitur tersebut penting karena merupakan solusi dari penyelesaian masalah.
2. *Should Have* (S)
Should Have memiliki arti bahwa suatu fitur tidak memiliki prioritas utama dalam pengerjaan sistem, tetapi penting dan memiliki nilai bagi pengguna.
3. *Could Have* (Co)
Could have memiliki arti bahwa suatu fitur yang dapat meningkatkan pengguna sistem, namun jika tidak dapat diimplementasikan tidak berpengaruh pada proyek.
4. *Won't Have This Time* (W)
Won't Have This Time memiliki arti bahwa suatu fitur dapat diimplementasikan setelah pengembangan selesai.

Tabel 2.5 merupakan cara dalam menjelaskan fitur perangkat lunak.

Tabel 2.5 Fitur Sistem

Kode Fitur	Deskripsi	Prioritas
[Kode suatu fitur, sebagai tiap fitur]	[Deskripsi dari fitur]	[Prioritas fitur]

Sumber : Bittner & Spence (2002)

2. Kebutuhan fungsional

Kebutuhan fungsional dari sistem menggambarkan apa dan bagaimana aktivitas yang harus dilakukan oleh sistem. Kebutuhan ini bergantung pada tipe aplikasi yang akan dikembangkan, apa yang diinginkan oleh pengguna dari aplikasi yang dikembangkan, dan bagaimana pendekatan dari pemangku kepentingan ketika menentukan kebutuhan aplikasi. Ketika menjelaskan kebutuhan dari pemangku kepentingan atau pengguna, Kebutuhan fungsional adalah salah satu cara untuk menjelaskan fungsi yang diberikan kepada pengguna yang bisa dipahami (Sommerville, 2009).

3. Kebutuhan Non-Fungsional








Kebutuhan non-fungsional adalah kebutuhan yang tidak secara langsung berkaitan dengan layanan yang diberikan oleh sistem kepada pengguna. Kebutuhan ini berkaitan dengan keandalan sistem, waktu proses dari sistem, dan keamanan sistem. Kebutuhan non fungsional juga menjelaskan bagaimana sebuah sistem dikembangkan sesuai dengan keadaan atau kondisi eksternal dari sistem seperti kebutuhan sistem untuk merespon perangkat *input output*, bagaimana sistem bisa menampilkan data yang representatif dan sesuai dengan keinginan pengguna, dan bagaimana sistem bisa menyesuaikan dengan sistem lain (Sommerville, 2009).

2.6 Business Process Model Notation (BPMN)





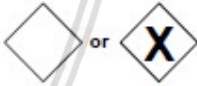

Untuk memodelkan proses bisnis pengajuan surat izin hingga penanganan selesai, maka dibutuhkan sebuah notasi yang dapat menggambarkan proses-proses bisnis tersebut. Sebuah model proses bisnis terdiri dari serangkaian model kegiatan dan *constraint* antara model-model kegiatan (Weske, 2012). BPMN merupakan singkatan dari *Business Process Modelling Notation*, yaitu suatu metodologi yang dikembangkan *Business Process Modelling Initiative* (BPMI) dalam memodelkan proses bisnis (Object Management Group, 2011). Tujuan dari BPMN adalah menyediakan notasi yang mudah dipahami oleh semua pengguna bisnis dan yang tidak kalah pentingnya adalah untuk memastikan bahwa bahasa XML yang dirancang untuk pelaksanaan proses bisnis dapat dinyatakan secara visual dengan notasi yang umum. Tidak seperti jenis diagram proses bisnis yang sebelumnya, BPMN telah ditambahkan notasi khusus untuk menggambarkan peristiwa berbasis pesan dan pesan lewat diantara organisasi. Tujuan utama dari BPMN adalah menyediakan suatu notasi standar yang mudah dipahami oleh semua pemangku kepentingan bisnis.

Berikut pada Tabel 2.6 merupakan beberapa penjelasan dari elemen-elemen yang digunakan dalam menggambarkan BPMN, antara lain :



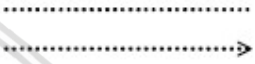
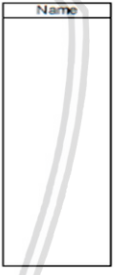
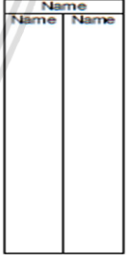


Tabel 2.6 Notasi pada BPMN

No	Tipe	Deskripsi	Simbol
1	<i>Event</i>	Menggambarkan suatu peristiwa yang terjadi selama berjalannya sebuah proses. <i>Event</i> memiliki pemicu atau hasil yang dapat berpengaruh pada aliran proses yang dimodelkan. Berdasarkan waktu dalam mempengaruhi aliran, <i>Event</i> dibagi menjadi 3 jenis , yaitu : <i>Start</i> , <i>Intermediate</i> dan <i>End</i> .	
	<i>Start</i> (Trigger : none)	Mengindikasikan letak sebuah proses atau koreografi akan dimulai. <i>Event</i> berupa <i>start</i> yang digambarkan dengan lingkaran kosong didalamnya menandakan bahwa <i>event</i> tersebut tidak memiliki pemicu.	
	<i>Start</i> (Trigger : message)	Sebuah pesan datang dari partisipan dan memicu mulainya proses.	
	<i>Intermediate</i>	Menggambarkan peristiwa yang terjadi diantara event jenis <i>start</i> dan <i>end</i> . <i>Event</i> ini akan mempengaruhi aliran proses, namun tidak akan memulai atau mengakhiri proses secara paksa.	
	<i>Intermediate</i> (Trigger : message catch)	Dapat digunakan baik untuk mengirim maupun menerima pesan. Dalam kolaborasi proses, notasi ini dapat digunakan untuk menghubungkan <i>event</i> antar partisipan yang terlibat.	
	<i>End</i>	Menggambarkan letak proses berakhir.	
2.	<i>Activity</i>	Merupakan istilah umum untuk pekerjaan yang dilakukan perusahaan dalam suatu proses. Notasi activity dapat digunakan untuk proses standar maupun dalam koreografi.	

Tabel 2.6 Notasi pada BPMN

No	Type	Deskripsi	Simbol
	<i>Manual task</i>	Merupakan jenis <i>Task</i> (tugas) yang dijalankan tanpa adanya proses bisnis mengeksekusi aplikasi atau mesin tertentu.	
	<i>User task</i>	Merupakan jenis aliran kerja <i>Task</i> (tugas) dimana manusia melakukan tugasnya dengan menggunakan bantuan dari sebuah aplikasi perangkat lunak dan dijadwalkan melalui sebuah daftar tugas.	
	<i>Service task</i>	Merupakan <i>Task</i> (tugas) yang menggunakan beberapa <i>service</i> singkat yang dapat berupa sebuah <i>web service</i> atau sebuah aplikasi otomatis.	
3.	<i>Gateway</i>	Digunakan untuk mengatur bagaimana aliran sekuen berinteraksi saat menyamakan dan memisahkan dalam sebuah proses. Sebuah <i>gateway</i> dapat memiliki beberapa masukan dan aliran keluaran.	
	<i>Exclusive Gateway</i>	<i>Diverging</i> pada <i>Exclusive Gateway</i> digunakan untuk membuat jalur alternatif dalam sebuah aliran proses. Pada penggunaannya sebagai <i>diverging</i> , hanya satu jalur yang dapat diambil. Sedangkan <i>covering</i> pada <i>Exclusive Gateway</i> digunakan untuk menggabungkan jalur alternatif. Masing-masing token aliran sekuen masuk diarahkan keluar tanpa sinkronisasi.	
	<i>Parallel Gateway</i>	Digunakan untuk menyinkronisasi aliran paralel dan menciptakan aliran paralel. <i>Parallel Gateway</i> membuat jalur paralel tanpa memeriksa kondisi apapun, setiap aliran sekuen keluar menerima token pada saat eksekusi. Saat aliran datang, <i>Parallel Gateway</i> akan menunggu aliran datang	

Tabel 2.6 Notasi pada BPMN

No	Tipe	Deskripsi	Simbol
		sebelum menggerakkan aliran melalui aliran sekuen keluar.	
4.	<i>Sequence Flow</i>	Digunakan untuk menunjukkan urutan yang aktivitasnya akan dijalankan dalam sebuah proses. Setiap aliran sekuen hanya memiliki satu sumber dan hanya satu target.	
5.	<i>Message Flow</i>	Digunakan untuk menunjukkan aliran pesan antara dua partisipan yang disiapkan untuk mengirim dan menerima pesan.	
6.	<i>Association</i>	Digunakan untuk menghubungkan informasi dan artefak dalam BPMN. Anak panah pada asosiasi mengindikasikan tujuan dari aliran, jika dibutuhkan.	
7.	<i>Pool</i>	Digunakan untuk merepresentasikan partisipan dalam sebuah kolaborasi. Dapat pula dianggap sebagai <i>swimlane</i> dan wadah grafis untuk memisahkan sekumpulan aktivitas dari <i>pool</i> lain.	
8.	<i>Lane</i>	Merupakan sebuah sub partisi dalam sebuah proses, biasanya pada sebuah <i>pool</i> dan akan memperbesar seluruh Panjang proses.	
9.	<i>Data Object</i>	Menyediakan informasi tentang hal yang dibutuhkan aktivitas untuk menjalankan dan/atau yang dihasilkan dari aktivitas tersebut.	
	<i>Data Store</i>	Menyediakan mekanisme aktivitas untuk mengambil atau memperbarui informasi tersimpan, yang akan berlangsung lama melampaui jangkauan proses. <i>Data Store</i> yang sama	

Tabel 2.6 Notasi pada BPMN

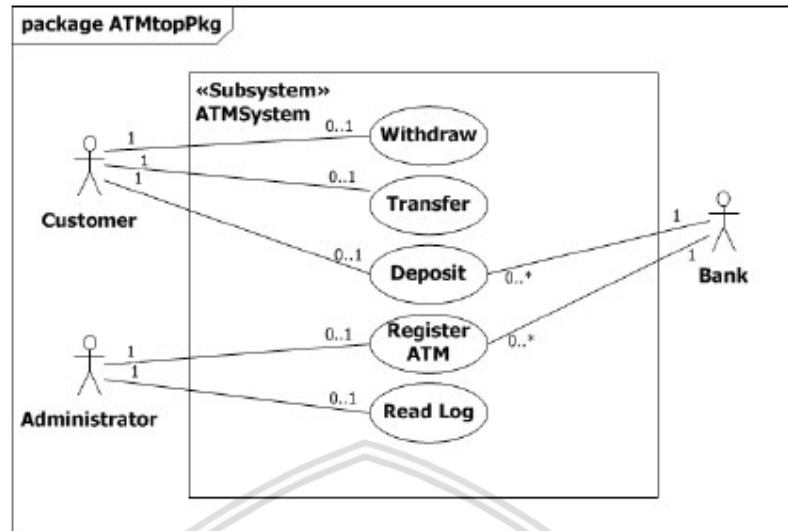
No	Tipe	Deskripsi	Simbol
		dapat divisualisasi melalui sebuah data store reference pada satu atau lebih tempat di dalam proses.	

2.7 Unified Modelling Language (UML)

Desain dan implementasi pada perangkat lunak merupakan aktivitas yang saling terkait, sehingga ketika melakukan pengembangan desain, *developer* perlu memperhatikan hal-hal tentang pengimplementasian dari desain yang dibuat. Hal ini juga menjadi pertimbangan pemilihan bahasa pemodelan yang tepat untuk menggambarkan perangkat lunak yang dibangun. Pada kasus pengembangan perangkat lunak dengan pendekatan *object-oriented*, UML tepat dipilih untuk mendokumentasikan desain yang akan dibuat (Sommerville, 2009). Saat ini, secara *de facto* *Unified Modeling Language* (UML) yang mulai diperkenalkan pada tahun 1997, telah berkembang menjadi bahasa standar pemodelan perangkat lunak (Kurniawan, 2018). Lebih lengkapnya UML dapat digunakan untuk menggambarkan, membangun, dan mendokumentasikan artifak yang diperlukan dalam proses pengembangan sistem. UML menyediakan beberapa pilihan fitur yang dapat digunakan untuk menggambarkan aspek-aspek penting dari sebuah sistem. Namun dalam penggunaannya, tidak diwajibkan menggunakan seluruh diagram yang ada, perancang sistem dapat memilih sesuai dengan kebutuhannya (Pressman, 2010). Pada poin pembahasan 2.6.1 hingga 2.6.4 berikut adalah diagram UML yang digunakan pada pengembangan SINKES.

2.7.1 Use case diagram

Use case diagram merupakan bagian UML yang menjelaskan aspek perilaku (*behavioral aspect*) dari sebuah sistem. Aspek perilaku dapat menjelaskan perubahan yang terjadi pada sistem seiring dengan perubahan waktu. Penjelasan dari setiap *usecase* pada diagram dijelaskan secara rinci dan terpisah pada *usecase scenario* (Kurniawan, 2018). *Use case diagram* adalah diagram yang menggambarkan kegunaan atau fungsi apa saja yang bisa dilakukan oleh sistem dengan menunjukkan bagaimana interaksi antara pengguna dan sistem terkait fungsi apa saja yang bisa digunakan oleh pengguna (Booch, et al., 2007). Maka dari itu, sudut pandang yang digunakan dalam membuat *use case* adalah sudut pandang dari sisi pengguna sistem (aktor) dan penamaannya harus didasari oleh tujuan yang ingin dicapai oleh aktor tersebut, dengan kata lain penamaan *use case* harus menggunakan kata kerja aktif (Kurniawan, 2018). *Use case* membuktikan bahwa jika digunakan dengan tepat, maka dapat menjadi nilai yang berharga sebagai bagian analisa kebutuhan pada proses pengembangan perangkat lunak. Keakuratan dalam penyusunan kebutuhanpun dapat dilakukan dengan lebih mudah dan tepat oleh tim pengembang dan stakeholder (Bittner & Spence, 2002).






Gambar 2.3 Use case diagram untuk sistem music



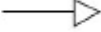
Sumber : Object Management Group (2017)

Menurut Kurniawan (2018) dalam mendesain *usecase diagram*, tidak dibenarkan melakukan abstraksi aktivitas CRUD (*create, read, update, delete*) kedalam 1 *use case*. Misalkan terdapat obyek “data pelanggan”, maka tidak dibenarkan membuat *use case* “Mengelola data pelanggan”. Hal ini dikarenakan ada ketidakjelasan fungsional utama dan hasil akhir dari *use case* tersebut. Maka dari itu, seharusnya *use case* yang dibuat adalah “Menambah data pelanggan”, “Melihat data pelanggan”, “Mengubah data pelanggan”, dan “Menghapus data pelanggan”. Didalam diagram *use case* terdapat 5 elemen penyusun dari diagram *use case* diantaranya:

Tabel 2.7 Simbol use case diagram

No	Simbol	Deskripsi
1	<div>Use case</div> 	Use case merupakan spesifikasi yang dilakukan oleh sistem
2	<div>Aktor</div> 	Merupakan tipe peran yang dijalankan oleh sebuah entitas untuk berinteraksi dengan sistem. Aktor dapat direpresentasikan oleh manusia, perangkat keras eksternal atau yang lain
3	<div>Association</div> 	Association merupakan sebuah koneksi antara aktor dan use case. Association mengindikasikan bahwa aktor dapat melakukan suatu use case. Dalam UML, association berarti bahwa sebuah aktor terlibat dalam sebuah use case.

Tabel 2.7 Simbol *use case diagram*

No	Simbol	Deskripsi
4	<i>Extend</i> «extends» 	<i>Extend</i> merupakan sebuah hubungan dimana memungkinkan kita untuk memodifikasi tingkah laku sebuah <i>use case</i> dasar dengan menambahkan sebuah <i>use case</i> tambahan.
5	<i>Include</i> «includes» 	<i>Include</i> menunjukkan bahwa <i>behaviour</i> sebuah <i>included use case</i> termasuk bagian dari sebuah <i>use case</i> dasar. Tujuannya untuk penggunaan kembali sebuah <i>use case</i> yang digunakan berulang kali dalam <i>use case</i> lain.
6	<i>Generalization</i> 	<i>Generalization</i> merupakan notasi hubungan antar aktor yang menghubungkan aktor yang lebih spesifik dengan aktor yang umum.

Sumber : (Sukanto & Shalahuddin, 2016)

2.7.2 *Use case scenario*

Use case scenario adalah bentuk *textual* yang menjelaskan secara detail interaksi antara aktor dan sistem dari sebuah *use case* (Kurniawan, 2018). *Use case scenario* menyediakan substansi model *use case* dan merupakan dasar untuk sebagian besar pemodelan *use case*. Representasi grafis pada *use case diagram* berguna dalam menggambarkan ruang lingkup sistem yang akan dibuat, namun tidak secara jelas menggambarkan apa yang dilakukan aktor dan umpan balik dari sistem. Maka dari itu, *use case scenario* dapat digunakan untuk mendeskripsikan secara lengkap tentang suatu *use case* (Bittner & Spence, 2002). Isi dari *use case scenario* ditunjukkan oleh Tabel 2.8.

Tabel 2.8 *Use case scenario*

<i>Brief description</i>	Merupakan penjelasan <i>use case</i> secara singkat.
<i>Actor</i>	Merupakan aktor pemilik <i>use case</i> yang berinteraksi secara langsung dengan sistem.
<i>Pre-condition</i>	Merupakan kondisi atau keadaan yang harus dipenuhi oleh sistem sebelum <i>use case</i> dapat dijalankan.
<i>Post-condition</i>	Merupakan kondisi atau keadaan sistem setelah sebuah <i>use case</i> selesai dijalankan.
<i>Extension</i>	Merupakan informasi terkait <i>use case</i> lain yang bersifat opsional, yang disertakan pada sebuah <i>use case</i> .
<i>Basic flow</i>	Merupakan penjelasan jalannya sebuah <i>use case</i> dengan alur normal.
<i>Alternative flow</i>	Merupakan alur opsional yang akan dijalankan ketika <i>use case</i> tidak berjalan sesuai dengan alur normal.

Tabel 2.8 Use case scenario

<i>Subflow</i>	Merupakan penyederhanaan dari beberapa kelompok alur <i>use case</i> yang terlalu detail.
----------------	---

Sumber : Diadaptasi dari Bittner & Spence (2002)

Pada penelitian ini, tabel use case scenario mengalami penyesuaian untuk memperjelas informasi yang disertakan. Adapun contoh tabel yang telah mengalami penyesuaian beserta penjelasannya ditunjukkan pada Tabel 2.9.

Tabel 2.9 Tabel use case scenario yang telah mengalami penyesuaian

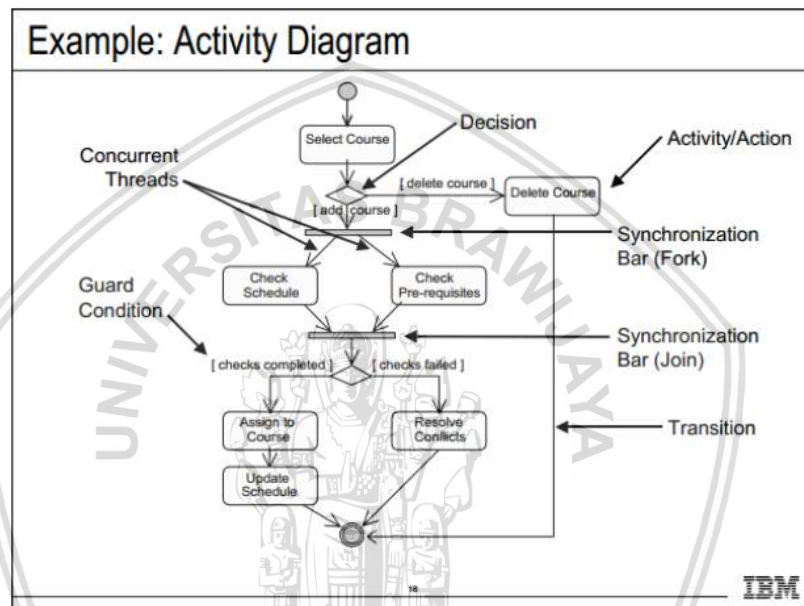
Kode Use Case	Berisi kode penanda yang diberikan pada tiap <i>use case</i> untuk mempermudah peruntutan.
Kode Fungsional	Berisi kode fungsional dari <i>use case</i> yang akan dijelaskan.
Nama	Berisi nama <i>use case</i> yang akan dijelaskan.
Aktor	Nama aktor pemilik <i>use case</i> .
Deskripsi	Berisi penjelasan mengenai <i>use case</i> .
Tujuan	Berisi tentang tujuan yang diharapkan dapat diraih oleh <i>use case</i> .
Kondisi Awal	Berisi prasyarat yang harus dipenuhi sebelum menjalankan <i>use case</i> .
Skenario Utama	
Berisi mengenai urutan alur interaksi antara aktor dan sistem dalam kondisi normal.	
Skenario Alternatif	
Berisi mengenai urutan alur interaksi antara aktor dan sistem jika kondisi normal tidak terpenuhi.	
Kondisi Akhir	Berisi kondisi akhir setelah alur interaksi dijalankan.
Ekstensi	Berisi informasi terkait <i>use case</i> lain yang bersifat opsional, yang disertakan pada sebuah <i>use case</i> .
Sub Aliran	Berisi penyederhanaan dari beberapa kelompok alur <i>use case</i> yang terlalu detail.

2.7.3 Activity diagram

Activity diagram adalah sebuah diagram yang menggambarkan alur dari aktivitas sistem, bisnis, alur kerja, atau proses yang ada pada sistem. *Activity Diagram* berfokus pada apa saja aktivitas berurutan yang harus dilakukan dan

siapa yang bertanggung jawab pada proses tersebut (Booch, et al., 2007). *Activity Diagram* sesuai dengan gambar 2.4 digunakan untuk mendefinisikan hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya
4. Rancangan menu yang ditampilkan pada perangkat lunak.



Gambar 2.4 Contoh *Activity Diagram*





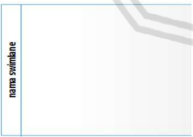
Sumber : IBM (2004)

Berikut merupakan penjelasan dari elemen – elemen tersebut.

1. *Decision* : Melakukan evaluasi kondisi yang ditentukan oleh *guard condition*.
2. *Concurrent Threads* : Proses yang berjalan bersamaan.
3. *Activity / Action* : Merepresentasikan langkah – langkah dari alur.
4. *Synchronization Bar* : Menunjukkan alur yang paralel, untuk *fork* merupakan awal percabangan, sedangkan *join* merupakan penggabungan proses.
5. *Guard Condition* : Pilihan kondisi yang dapat dilakukan.
6. *Transition* : Menunjukkan proses yang akan berjalan selanjutnya

Berikut pada tabel 2.8 adalah simbol-simbol yang digunakan dalam diagram aktifitas (Booch, 2005):

Tabel 2.10 Simbol-simbol *activity diagram*

No.	Simbol	Deskripsi
1.	Status awal (<i>start</i>) 	Status awal aktivitas sistem
2.	Aktivitas 	Proses yang dilakukan oleh aktor atau objek. Biasanya diawali dengan kata kerja
3.	<i>Branch</i> /cabang 	Menggambarkan cabang yang dimungkinkan dalam sebuah alur aktifitas. Sebuah cabang mungkin memiliki satu arus masuk dan dua atau lebih arus keluar dan digambarkan dalam simbol <i>diamond</i> . Pada setiap aliran keluar, ditempatkan ekspresi <i>Boolean</i> , yang dievaluasi saat memasuki cabang. Pada kondisi tersebut cabang berperan sebagai <i>decision</i> . Ketika dua aliran kontrol kembali bergabung bersama-sama, simbol <i>diamond</i> dapat digunakan dengan dua arus masuk dan satu arus keluar. Pada kondisi ini cabang berperan sebagai <i>merge</i> .
4.	Status akhir 	Status akhir yang dilakukan sistem, menandakan bahwa alur proses telah berakhir
5.	<i>Swimlane</i> 	Bertujuan untuk partisi aktivitas pada diagram aktivitas dalam kelompok-kelompok, setiap kelompok mewakili organisasi bisnis yang berperan dalam sistem.


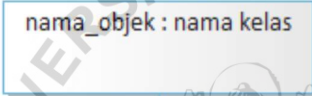

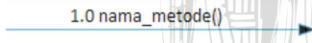

2.7.4 Sequence diagram

Sequence diagram atau diagram sekuen menggambarkan interaksi antar objek didalam dan di sekitar sistem (termasuk pengguna, tampilan, dan sebagainya) berupa pesan atau *message* yang digambarkan terhadap waktu. Diagram sekuen biasa menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari *event* untuk menghasilkan keluaran tertentu. *Message* yang digambarkan pada diagram sekuen nantinya akan dipetakan menjadi



fungsi/metoda dari *class* pada tahapan desain selanjutnya. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan simbol khusus untuk objek *boundary*, *controller* dan *persistent entity* (Yasin, 2012).

Diagram sekuen merupakan diagram interaksi yang menekankan pada saat permintaan pesan yang menunjukkan serangkaian peranan dan pesan yang dikirim dan diterima oleh bagian yang ada pada peranan tersebut. Diagram sekuen digunakan untuk menggambarkan *dynamic view* dari sistem (Booch, et al., 2007). Simbol-simbol yang digunakan pada diagram *sequence* ditunjukkan pada Tabel 2.9

Tabel 2.11 Simbol-simbol *sequence diagram*

Nama	Simbol	Deskripsi
<i>Lifeline</i>		Menunjukkan keberadaan objek selama periode waktu.
Objek		Menyatakan objek yang berinteraksi menggunakan pesan atau <i>message</i> .
<i>Activation Bar</i>		Menunjukkan periode waktu selama sebuah objek melakukan kegiatan, baik secara langsung maupun melalui prosedur perintah.
Pesan tipe call (synchronous)		Menyatakan suatu objek memanggil fungsi yang ada pada objek lain atau dirinya sendiri. Arah panah mengarah pada objek yang memiliki fungsi, karena ini memanggil sebuah fungsi maka fungsi yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
Pesan tipe send (asynchronous)		Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya

Tabel 2.11 Simbol-simbol *sequence diagram* (lanjutan)





Nama	Simbol	Deskripsi
Pesan tipe <i>return</i>		Menyatakan bahwa suatu objek yang telah menjalankan suatu fungsi menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
Pesan tipe <i>destroy</i>		Objek dapat diakhiri selama interaksi yang berlangsung. <i>Lifeline</i> objek berakhir apabila menerima pesan <i>destroy</i> . Bila ada pesan tipe <i>create</i> maka sebaiknya ada pesan tipe <i>destroy</i> .

Sumber : Booch, et al., (2007)


2.7.5 Class diagram

Class diagram digunakan untuk menunjukan apa saja *class* atau objek yang berkaitan dengan system dan hubungan antara *class* untuk saling bertukar informasi seperti apa. Suatu *class diagram* merepresentasikan struktur dari sebuah *class* yang ada di dalam sistem (Booch, et al., 2007). Elemen yang paling penting dalam sebuah *class diagram* adalah *class* dan hubungannya. Tabel 2.10 menjelaskan simbol dari *class diagram*.

Tabel 2.12 Simbol *Class Diagram*

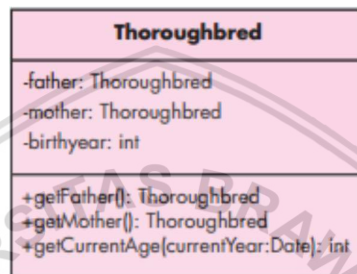
No	Simbol	Deskripsi
1	Class 	<i>Class</i> merupakan sebuah <i>template</i> dimana objek dibuat. <i>Class</i> mendefinisikan atribut, operasi dan <i>instance</i> .
2	<i>Association</i> 	Relasi yang menunjukan hubungan antara dua <i>class</i> .
3	<i>Generalization</i> 	Relasi yang menunjukan pewarisan dari <i>class</i> utama (<i>parent</i>) ke <i>class</i> anak (<i>child</i>)
4	<i>Dependency</i> 	Relasi yang menunjukan suatu <i>class</i> bergantung pada <i>class</i> yang lain

Tabel 2.12 Simbol *Class Diagram*

No	Simbol	Deskripsi
5	Agregasi / <i>Aggregation</i> 	Relasi yang menunjukkan sebuah elemen yang terdiri dari beberapa komponen kecil

Sumber : (Sukamto & Shalahuddin, 2016)

Berikut pada Gambar 2.5 merupakan contoh dari penggambaran *class diagram* beserta dengan atribut, fungsi serta jenis fungsi-fungsi yang dapat dilakukan oleh kelas Thoroughbred:

**Gambar 2.5 Class diagram dari kelas Thoroughbred**

Sumber : Pressman (2010)

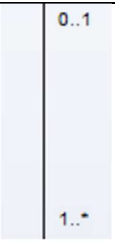

Kotak teratas dari Gambar 2.5 berisi nama dari *class* yang pada umumnya mewakili sebuah obyek yang ada pada sistem. Kotak dibawahnya berisikan penjelasan nama dan tipe data dari atribut-atribut yang dimiliki oleh obyek kelas tersebut. Sedangkan kotak terakhir yang paling bawah berisikan segala macam fungsi yang dimiliki dan dapat dilakukan oleh obyek dari instansiasi kelas tersebut serta menjelaskan tipe data keluaran dari fungsi tersebut. Atribut dan fungsi memiliki beberapa tipe hak akses enkapsulasi yang dinotasikan sebagai +, ~, #, - secara urut mewakili public, package, protected, dan private (Pressman, 2010)

2.8 Conceptual Data Model (CDM)

Conceptual Data Model (CDM) mengidentifikasi relasi level tertinggi antara entitas-entitas yang berbeda dengan pandangan tentang pemakaian data dalam sistem. Dalam CDM, konsep data dimodelkan dengan mengikutsertakan entitas-entitas penting dan relasi yang ada diantara mereka, tanpa menyertakan atribut dan *primary key* (Sukamto & Shalahuddin, 2016). CDM membentuk konsep dasar dan semantik dari domain tertentu dan membantu mengkomunikasikannya kepada banyak pemangku kepentingan (Sparx Systems, 2017). Pada penelitian ini, CDM akan digunakan untuk menggambarkan tabel dan hubungan antar tabel yang akan dibangun sebagai media penyimpanan data SINKES. Berikut adalah simbol-simbol yang terdapat pada CDM menurut (Sukamto & Shalahuddin, 2016).

Tabel 2.13 Simbol-simbol *Conceptual Data Model* (CDM)

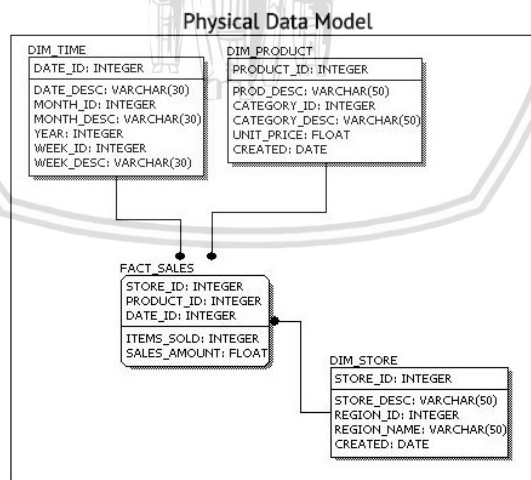
Simbol	Deskripsi
--------	-----------

	Relasi menggambarkan hubungan antar tabel yang terdiri atas nama relasi dan <i>multiplicity</i> .
	Entitas atau tabel menggambarkan tempat menyimpan data dalam sistem basis data yang terdapat dalam sistem informasi.

2.9 Physical Data Model (PDM)

Physical Data Model (PDM) secara visual merepresentasikan struktur data seperti yang diimplementasikan oleh skema relasi basisdata (Sparx Systems, 2017). CDM merupakan pemodelan basis data pada tingkatan paling detail dari sebuah desain basis data. Diagram PDM menggambarkan secara keseluruhan bentuk fisik dari tabel seperti nama kolom, panjang data dari kolom tersebut, *primary key* serta tipe *constraint* lain seperti foreign key untuk menggambarkan relasi antara satu tabel dengan tabel yang lain. Diagram PDM dibuat berdasarkan tipe dari perangkat lunak pendukung database yang digunakan (*Physical Data Model*, 2001).

Berikut pada gambar 2.6 adalah contoh dari Diagram PDM



Gambar 2.6 Contoh Diagram *Physical Data Model*
(Sumber: *Physical Data Model*, 2001)

Pada gambar 2.6 adalah contoh dari diagram PDM untuk menyimpan data penjualan dari sebuah toko. Pada diagram tersebut entitas dari tabel fakta penjualan sudah terlihat jelas bentuk fisik dari tabelnya yang digambarkan dengan

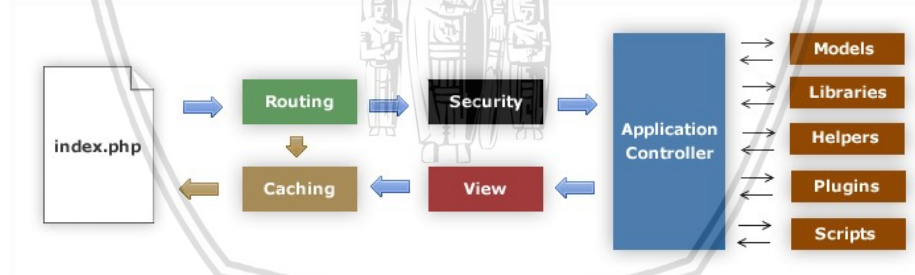
tabel FACT_SALES. Tabel FACT_SALES juga memiliki atribut *foreign key* dari tabel DIM_PRODUCT pada kolom PRODUCT_ID, DIM_TIME pada kolom DATE_ID, dan DIM_STORE pada kolom STORE_ID karena tabel FACT_SALES memiliki relasi dengan ketiga tabel tersebut. Pada diagram tersebut juga sudah terlihat tipe data yang digunakan pada kolom dan berapa panjang data yang dibutuhkan pada kolom tersebut.

2.10 Pola Model-View-Controller

Pada konsep MVC, bagian Model dikerjakan oleh *programmer* sekaligus *database analyst*, dimana kode-kode di bagian ini merupakan kode struktur data yang berisi koneksi dan fungsi di dalam pengolahan database. Sedangkan View dikerjakan oleh designer html, css, dan js karena kode di bagian ini untuk menampilkan tampilan website yang menarik dan interaktif. Terakhir, *Controller* dikerjakan oleh programmer inti PHP karena menangani hubungan Model dengan View, algoritma, *logic*, HttpRequest, dan lain sebagainya (Hustinawati, et al., 2014).

2.10.1 Framework CodeIgniter

Codeigniter adalah aplikasi *open source* yang berupa *framework* dengan menggunakan konsep model MVC untuk memisahkan antara logik dari aplikasi (*Controller*) dengan tampilan aplikasi (*View*) dan database aplikasi (*Model*). Codeigniter memudahkan developer untuk membuat aplikasi *website* dengan cepat dan mudah dibandingkan dengan membuatnya dari awal (Hustinawati, et al., 2014).



Gambar 2.7 Alur kerja Codeigniter

Sumber: Dokumentasi Codeigniter

Dari gambar 2.5, secara sederhana cara kerja Codeigniter dimulai ketika `index.php` menginisialisasi kebutuhan untuk menjalankan codeigniter. Data dari `index.php` lewat melalui `routing`, dimana pada `routing` ini akan ditentukan kegiatan apa yang diminta oleh *client*. Jika *cache* aktif dan ternyata data yang diminta sudah pernah diakses sebelumnya, maka alur utama diabaikan dan *request* dari *client* didapatkan dari *cache* yang sudah ada. Pada *security*, data akan diproses dan dianalisa keamanannya. Pada *controller*, *request* diproses sesuai jenis *request*, seperti *driver*, *model*, *library*, *helper*, dan lain-lain. Setelah diproses, *view* menampilkan informasi yang siap ditampilkan di web browser (Widodo, 2015).

2.11 Pengujian

Pengujian perangkat lunak mengacu pada proses evaluasi perangkat lunak dengan maksud untuk mengetahui kesalahan di dalamnya. Pengujian perangkat lunak adalah teknik yang ditujukan untuk mengevaluasi atribut atau kemampuan suatu program menentukan bahwa program tersebut memenuhi kualitasnya (Sawant, et al., 2012). Dalam istilah yang paling sederhana, pengujian berarti mengamati eksekusi sistem perangkat lunak untuk memvalidasi apakah berperilaku seperti sesuai yang dirancang dan mengidentifikasi malfungsi potensial (Bertolino, 2007).

Pengujian *black-box* adalah pengujian perangkat lunak berdasarkan persyaratan keluaran dan tanpa pengetahuan tentang struktur internal atau pengkodean dalam program. Tujuannya adalah untuk menguji seberapa baik komponen sesuai dengan persyaratan yang diterbitkan untuk komponen tersebut. Pengujian *black-box* sedikit atau tidak berkaitan dengan struktur logis internal sistem, hanya memeriksa aspek fundamental sistem. Ini memastikan bahwa *input* diterima dengan benar dan *output* diproduksi dengan benar (Sawant, et al., 2012).

2.11.1 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan pengujian perangkat lunak berhubungan dengan kebutuhan pengguna, *requirement*, dan proses bisnis yang dilakukan untuk menentukan apakah sistem telah memenuhi kriteria untuk diterima oleh pengguna dan *stakeholder* lain yang terlibat dalam pembangunannya. Tujuan utama dilakukannya UAT adalah untuk memastikan apakah suatu sistem yang baru telah memenuhi persyaratan dan kebutuhan dari suatu bisnis serta mampu melakukan apa yang telah ditentukan sebelumnya (Hambling & Goethem, 2013).

Adapun alasan-alasan penggunaan UAT dalam pembangunan sebuah sistem informasi menurut (Hambling & Goethem, 2013) adalah sebagai berikut.

1. Manajemen resiko (Menghindari kegagalan besar)

Dalam pembangunan sistem informasi, kegagalan merupakan resiko yang sulit dihindari meskipun sistem yang dibangun telah dianggap sangat baik. Dengan menggunakan UAT, maka potensi kegagalan dapat diketahui sedini mungkin.

2. Membangun kepercayaan dalam mencapai manfaat bisnis

Untuk memastikan bahwa pengguna perangkat lunak akan mendapatkan produktivitas yang diharapkan atau keuntungan lain, maka perlu menempatkan sistem untuk digunakan pada lingkungan yang sebenarnya dan mencoba menggunakannya melalui beberapa contoh situasi yang diharapkan dapat menambah nilai. Jenis pengujian ini akan didasarkan pada pemahaman tentang bagaimana sistem akan digunakan untuk mendapatkan manfaat bisnis dan menyiapkan kasus uji yang realistis untuk melihat bahwa ia mampu memberikan manfaat tersebut.

3. Penilaian (Mendapatkan proses bisnis yang benar)

Pengguna perlu memeriksa perilaku sistem sebagai bagian dari keseluruhan proses bisnis, menggunakannya dengan cara yang telah disepakati. Salah satu manfaat lebih lanjut dari UAT adalah bahwa tindakan menentukan kesesuaian sistem untuk tujuan tentu saja melibatkan perbandingan perilaku sistem dengan harapan pengguna. Jika perbandingan dilakukan secara sistematis dan formal, maka hasil utama dari UAT mungkin adalah menerima atau tidak menerima sistem.

4. Persiapan (Menilai kesiapan layanan)

Sebelum perangkat lunak benar-benar dilepaskan kedalam bisnis dan bahwa bisnis, perlu dilakukan penilaian kesiapan. Aspek kesiapan ini adalah tentang seberapa kuat sistem dalam beroperasi, seberapa dekat dengan kesesuaian yang diharapkan.

UAT merupakan pengujian perangkat lunak yang bergantung kepada keterlibatan pengguna. Hal ini disebabkan karena dibutuhkan sudut pandang yang unik dari pengguna untuk mengetahui kesesuaian penggunaan suatu perangkat lunak dengan konteks bisnis yang telah dirancang. Untuk keperluan tersebut, maka dalam melakukan UAT diperlukan keterlibatan *stakeholder* dari perangkat lunak yang dibangun. *Stakeholder* yang biasanya terlibat dalam UAT tersebut antara lain adalah:

1. *Sponsor*

Merupakan pihak baik perorangan maupun kelompok, yang menugaskan dan mendefinisikan maksud bisnis. Dalam kasus lain juga termasuk pihak yang memiliki anggaran untuk mengimplementasikan sistem.

2. *Business Manager*

Merupakan pihak yang bertanggung jawab untuk memastikan dan memberikan manfaat bisnis yang diharapkan dari implementasi sistem.

3. *End-User*

Merupakan pihak yang akan menggunakan dan mengoperasikan sistem dalam kesehariannya.

4. *Developer*

Merupakan pihak yang bertanggung jawab terhadap implementasi dari sistem dan yang akan menunjang upaya proses pengujian.

Pada penelitian ini, proses pelaksanaan UAT sendiri akan dilakukan dengan memberikan kuesioner yang berisi pertanyaan untuk pengujian yang berhubungan dengan kebutuhan pengguna dan *requirement* sistem kepada *stakeholder* jenis *business manager*. Kuesioner tersebut akan disajikan dengan beberapa pilihan jawaban persetujuan dari pelanggan menggunakan skala *likert*. Dalam melakukan UAT, terdapat beberapa urutan proses yang harus dilakukan menurut (Hambling & Goethem, 2013), yaitu.

1. Perencanaan (*Planning*)

Pada bagian perencanaan, akan dilakukan penguraian strategi UAT seperti penentuan terhadap hal-hal yang ingin dicapai, kriteria penerimaan, tujuan melakukan UAT, kriteria masukan dan menentukan pengujian yang akan dilakukan.

2. Desain pengujian (*Test Design*)

Desain pengujian merupakan proses merubah tujuan umum pengujian menjadi kondisi uji *tangible* dan kasus uji. Aktivitas yang dilakukan diantaranya menentukan hirarki desain uji, identifikasi kondisi uji, desain kasus uji, desain naskah uji dan pembuatan data.

3. Implementasi pengujian

Ketika desain pengujian telah dilaksanakan, maka selanjutnya adalah melakukan implementasi pengujian yang telah disusun sedemikian rupa. Dalam proses implementasi ini, terdapat aktivitas seperti menentukan jadwal pengujian, implementasi jadwal pengujian, identifikasi progress, menyusun laporan status, dan membuat ringkasan setelah pengujian dilakukan.

4. Evaluasi sistem

Hasil yang telah didapatkan pada tahap implementasi selanjutnya dijadikan dasar dalam melakukan analisis hasil akhir terkait diterima atau tidaknya sistem yang telah dibangun.

2.11.1.1 Kriteria penerimaan UAT

Kriteria penerimaan (*acceptance criteria*) digunakan sebagai acuan dalam menilai kesesuaian sistem sebelum dirilis pada pengguna. Kriteria penerimaan yang tidak akurat atau lengkap, maka akan mengarahkan pada hasil pengujian yang memiliki tingkat kepuasan rendah (Hambling & Goethem, 2013). Menurut (Naik & Tripathy, 2008), berikut beberapa kriteria penerimaan yang dapat digunakan.

Tabel 2.14 Kriteria Penerimaan UAT

Kriteria	Deskripsi
<i>Confidentially and Availability</i>	<p><i>Confidentiality</i> adalah satu set aturan yang membatasi hak akses terhadap informasi.</p> <p><i>Availability</i> adalah jaminan akses yang dapat dipercaya terhadap informasi oleh orang-orang yang berwenang.</p>
<i>Backup and recovery</i>	Memiliki fokus terhadap pencadangan dan pemulihan data.
<i>Competitive Edge</i>	Memiliki fokus terhadap kelebihan sistem yang lebih inovatif .

Tabel 2.14 Kriteria Penerimaan UAT

Kriteria	Deskripsi
<i>Data Conversion</i>	Memiliki fokus terhadap kemampuan sistem dalam menjaga data saat terjadi perubahan jenis atau bentuk.
<i>Data Integrity</i>	Memiliki fokus terhadap kemampuan sistem dalam melakukan pengolahan data.
<i>Functional Correctness and Completeness</i>	Memiliki fokus terhadap kelengkapan implementasi spesifikasi kebutuhan yang telah dideskripsikan sebelum pembangunan sistem.
<i>Usability</i>	Memiliki fokus pada tingkat kemudahan penggunaan sistem.
<i>Performance</i>	Memiliki fokus terhadap kinerja sistem.

2.11.1.2 Desain kasus uji

Kasus uji merupakan sebuah dokumen yang memiliki serangkaian data uji, pra kondisi, hasil yang diharapkan dan kondisi setelah pengujian, dikembangkan untuk skenario pengujian dengan tujuan memverifikasi pemenuhan kebutuhan yang spesifik. Berikut adalah contoh dokumentasi kasus uji untuk melakukan pengujian validasi fungsional sistem guna memastikan bahwa fungsional telah diimplementasikan dengan sesuai.

Test step	Test description	Expected results	Pass ✓/Fail ✗
1.	From the Excelsior home page select the Contracts module	An error page should be displayed 'Access denied. You do not have permission to perform this action or access this resource. You can request access below' A 'Request Access' section on the screen will also be displayed	
2.	Close the error page displayed by selecting the X button in the top right corner of the screen	The home page should be displayed	

Gambar 2.8 Contoh kasus uji validasi

Sumber : Hambling & Goethem (2013)

Dari contoh pembuatan kasus uji seperti yang ditunjukkan pada Gambar 2.10, dilakukan penyesuaian dengan menambahkan beberapa poin seperti Kode Pengujian sebagai penanda, Kode *Use Case* untuk memperjelaskan *use case* yang sedang diujikan, Fungsional Sistem untuk menyebutkan nama fungsional yang diujikan, Hasil yang didapat untuk menjelaskan hasil pengujian yang didapatkan.

Berikut pada Tabel 2.14 merupakan Tabel kasus uji yang telah mengalami penyesuaian.

Tabel 2.15 Tabel rancangan kasus uji validasi yang mengalami penyesuaian

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang Diharapkan	Hasil yang Didapatkan	Status (Valid / Tidak Valid)

Setelah melakukan uji validasi sistem, maka selanjutnya adalah pengujian penerimaan pengguna berdasarkan kriteria penerimaan. Tabel 2.16 menunjukkan tabel desain kasus uji yang akan diujikan, sedangkan Tabel 2.17 menunjukkan tabel hasil dari kasus uji dengan penilaian menggunakan skala Likert. Penjelasan kolom masing-masing tabel berada didalam tabel tersebut.

Tabel 2.16 Tabel rancangan kasus uji penerimaan pengguna

No	Kriteria	Daftar Pertanyaan
Berisi nomor urut kasus uji.	Berisi kriteria penerimaan yang dijadikan dasar membuat pertanyaan.	Berisi pertanyaan kasus uji.

Tabel 2.17 Tabel hasil pengujian penerimaan pengguna

No.	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
Berisi nomor urut kasus uji.	Berisi pertanyaan kasus uji.	Kode penanda kasus uji.	Pilihan jawaban sangat tidak setuju.	Pilihan jawaban tidak setuju.	Pilihan jawaban netral.	Pilihan jawaban setuju.	Pilihan jawaban sangat setuju.

2.11.1.3 Likert scale

Skala *Likert* merupakan skala tanggapan psikometri yang sering digunakan dalam kuesioner untuk mendapatkan tingkat persetujuan dari responden menggunakan pernyataan atau serangkaian pernyataan (Bertram, 2016). Umumnya *skala likert* menggunakan skala yang terdiri dari 5 poin secara terurut dimulai dari “Sangat Tidak Setuju” di salah satu ujung kemudian “Sangat Setuju” di ujung lainnya kemudian “Antara Setuju maupun Tidak Setuju” di tengah-tengahnya. Terkadang juga digunakan skala 4 poin atau angka genap lainnya yang

bertujuan untuk menghasilkan ukuran ipsatif atau pilihan paksa. Setiap tingkat pada skala diberikan nilai numerik yang biasanya dimulai dari 1 dan ditambah satu untuk tiap tingkatnya. Berikut pada Gambar 2.10 merupakan contoh dari skala *likert* dengan menggunakan skala 5 poin.



Gambar 2.10 Contoh Skala *Likert*

Sumber : Bertram (2016)

Dalam implementasi skala *likert* terdapat beberapa pernyataan yang masing-masing memiliki bobot nilai tertentu. Proses perhitungan hasil dari skala *likert* diawali dengan memberikan bobot nilai pada tiap jawaban. Berikut pada Tabel 2.6 merupakan rincian bobot nilai dari jawaban. Sedangkan pada Tabel 2.7 merupakan persentase untuk setiap jawaban.

Tabel 2.6 Bobot Nilai Jawaban

A = Sangat Setuju	5
B = Setuju	4
C = Netral	3
D = Kurang	2
E = Sangat Kurang	1

Sumber: Choizes (2017)

Tabel 2.7 Persentase Nilai Jawaban

80 sampai 100%	Sangat Setuju
60 sampai 79,99%	Setuju
40 sampai 59,99%	Netral
20 sampai 39,99%	Tidak Setuju
0 sampai 19,99%	Sangat Kurang Setuju

Sumber: Choizes (2017)

Setelah bobot nilai dan persentase nilai telah ditentukan, selanjutnya adalah mengkalikan jumlah jawaban dengan bobot nilainya. Untuk perhitungannya akan dijelaskan pada Rumus 2.1 berikut:

1. Jawaban Sangat Setuju (A) $= n \times 5$
2. Jawaban Setuju (B) $= n \times 4$
3. Jawaban Netral (C) $= n \times 3$

4. Jawaban Tidak Setuju (D) = $n \times 2$

5. Jawaban Sangat Tidak Setuju (E) = $n \times 1$

$$\text{Total Nilai} = (n \times 5) + (n \times 4) + (n \times 3) + (n \times 2) + (n \times 1) \quad (2.1)$$

n merupakan jumlah dari responden yang menjawab. Perhitungan selanjutnya adalah mencari nilai terendah (Y) dan nilai tertinggi dari hasil skala *likert* tersebut. Rumus yang digunakan dijelaskan pada Rumus 2.2 dan Rumus 2.3, dimana $N1$ merupakan nilai tertinggi skala *likert*, $N2$ merupakan nilai terendah skala *likert*, n merupakan jumlah responden, dan U merupakan jumlah uji kasus.

$$Y = N1 \times n \times U \quad (2.2)$$

$$Y = N1 \times n \times U \quad (2.3)$$

Sedangkan untuk menghitung hasil dari UAT dengan menggunakan skala *likert* dijelaskan pada Rumus 2.4 berikut.

$$\text{Rumus Index \%} = (\text{Total Nilai} \div Y) \times 100 \quad (2.4)$$

2.11.1.4 Responsiveness testing

Responsiveness testing adalah pengujian non fungsional sistem untuk memeriksa apakah sistem dapat ditampilkan dengan baik pada berbagai ukuran resolusi layer perangkat. *Responsiveness Web Design* (RWD) merupakan jenis desain tampilan *website* yang memungkinkan bahwa *website* yang dikembangkan dapat tampil dengan baik pada berbagai jenis perangkat dan resolusi. Dengan RWD, jika pengguna beralih dari *desktop* ke tablet, maka *website* akan secara otomatis menyesuaikan perubahan resolusi seperti perubahan resolusi gambar, tabel dan berbagai komponen lainnya sesuai kemampuan perangkat masing-masing (Swati, 2018). Untuk menguji tingkat *responsiveness website* yang sedang dibangun, salah satunya dapat menggunakan *tools* berupa *website* pengujian *responsiveness* seperti yang tersedia pada <http://responsivedesignchecker.com/> dengan memasukkan *link* URL *website* yang sedang dikembangkan ke dalam *website* pengujian. SINKES yang akan dikembangkan pada penelitian ini diharapkan dapat diakses dengan menggunakan berbagai ukuran resolusi layer perangkat pengguna, maka dari itu pengujian *responsiveness* perlu dilakukan untuk memastikan bahwa tampilan SINKES dapat beradaptasi dengan baik di berbagai ukuran tersebut.



Gambar 2.9 Tampilan pada website <http://responsivedesignchecker.com/>

2.11.1.5 Compatibility testing

Sebelum merilis perangkat lunak untuk digunakan oleh pengguna sesungguhnya, maka *developer* wajib memastikan bahwa perangkat lunak dapat diakses dan berjalan dengan baik diberbagai lingkungan perangkat pengguna. Untuk itu, perlu dilakukan pengujian bernama *Compatibility testing*. *Compatibility testing* sendiri merupakan pengujian untuk memastikan bahwa sistem yang dibangun dapat bekerja dengan sama persis ketika diakses melalui bermacam *platform*, sistem operasi dan *Database Management System* (DBMS) (Naik & Tripathy, 2008). Salah satu alat bantu uji yang dapat digunakan adalah *Sortsite*. *Sortsite* merupakan alat bantu yang menguji *compatibility* sistem perangkat lunak pada komponen *tag* HTML, CSS, *script*, kesalahan dalam pemanggilan halaman, teknologi yang tidak didukung oleh *browser* tertentu, dan lain-lain (PowerMapper, 2017).

The screenshot shows the SortSite 'Compatibility' tab for the URL <http://before.powermapper.com/>. The interface includes tabs for Summary, Issues, and Pages. Under the Compatibility tab, there are sub-tabs for Errors, Accessibility, Compatibility, Privacy, Search, Standards, and Usability. A message states: 'This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.' Below this is a table showing compatibility results across various browsers and versions. The table has columns for Browser, Internet Explorer (versions 7.0, 8.0, 9.0, 10.0, 11.0), Edge (version 13), Firefox (versions ≤ 45, 46), Safari (versions ≤ 8.0, 9.0), Opera (versions ≤ 35, 36), Chrome (versions ≤ 48, 49), and iOS (versions ≤ 6.0, 8.0, 9.0). The rows represent Critical Issues, Major Issues, and Minor Issues. Critical Issues show two red dots for Internet Explorer 7.0 and 8.0, and blue checkmarks for all other browsers. Major and Minor Issues show blue checkmarks for all tested browsers. A footnote states: '* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original browser.' Below the table is a section for 'Priority 1' with the text '1 issues on 1 pages'.

Browser	Internet Explorer					Edge	Firefox		Safari		Opera		Chrome		iOS		
Version	7.0	8.0	9.0	10.0	11.0	13	≤ 45	46	≤ 8.0	9.0	≤ 35	36	≤ 48	49	≤ 6.0	8.0	9.0
Critical Issues	●	●	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original browser.

Priority 1
1 issues on 1 pages

Gambar 2.10 Contoh hasil *Compatibility Testing* menggunakan *Sortsite*

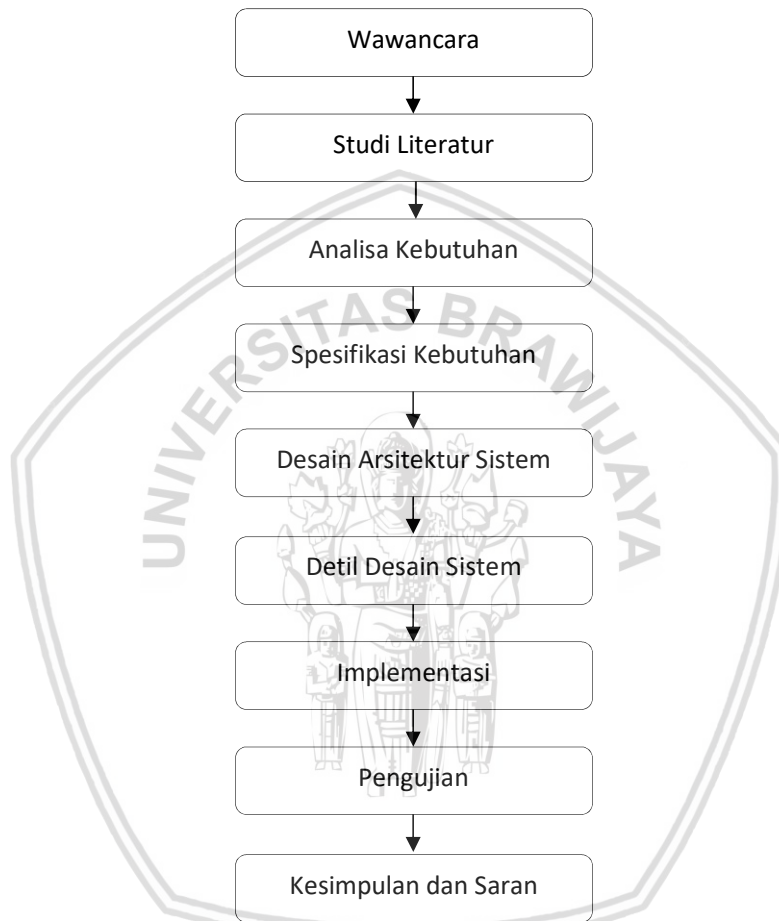
Sumber : (PowerMapper, 2017)



BAB 3 METODOLOGI

3.1 Metodologi penelitian

Metode penelitian yang akan dilakukan terdiri dari beberapa proses yang dilakukan secara beruntun, adapun diagram alir penelitian ini ditunjukkan pada Gambar 3.1 berikut.



Gambar 3.1 Diagram alir metodologi penelitian

3.1.1 Wawancara

Pada tahap ini, dilakukan proses pengumpulan data dengan menggunakan teknik wawancara tidak terstruktur. Penelitian ini merupakan bagian dari pengembangan sistem informasi yang dilakukan oleh CV. Sarana Utama Solusindo selaku pihak yang dipercaya Dinas Kesehatan Kota XYZ untuk mengembangkan sistem yang dapat membantu kinerja Dinas Kesehatan Kota XYZ dalam melayani masyarakat. Wawancara awal dilakukan dengan pihak CV. Sarana Utama Solusindo sebagai narasumber untuk mengetahui proses bisnis pengajuan izin kerja/praktik, *stakeholder* yang terlibat, permasalahan yang muncul terkait

penanganan perizinan di Dinas Kesehatan Kota XYZ dan dokumen pendukung sebagai dasar pengembangan sistem.

3.1.2 Studi literatur

Pada tahap studi literatur ini, dilakukan pencarian dan pengumpulan referensi dari jurnal, buku, artikel, *e-book* dan naskah penelitian untuk mendapatkan penjelasan teori maupun prinsip yang dijadikan dasar untuk mendukung penelitian. Adapun studi literatur yang dimaksud berkaitan dengan hal-hal berikut:

1. Penelitian sejenis yang pernah dilakukan
2. *Business Process Modeling*
3. *Unified Modeling Language* (UML)
4. Pola perancangan MVC
5. *Software Development Life Cycle* (SDLC) *V-Model*
6. Pengujian perangkat lunak

3.1.3 Analisis kebutuhan

Tahap analisis kebutuhan ini merupakan representasi dari *Requirement Analysis* pada SDLC *V-Model*. Data yang didapatkan dari hasil wawancara yang dilakukan pada tahap Wawancara digunakan sebagai dasar analisa kebutuhan sistem yang akan dikembangkan. Pada tahap ini akan dilakukan aktivitas seperti berikut.

1. Analisa proses bisnis yang sedang berjalan dan pemodelannya menggunakan *Business Process Notation Model* (BPMN).
2. Analisa permasalahan yang terjadi pada proses bisnis yang sedang berjalan.
3. Analisa proses bisnis pemodelan proses bisnis usulan untuk menyelesaikan permasalahan menggunakan *Business Process Notation Model* (BPMN).
4. Analisa kebutuhan sistem yang dirunut dari tipe pemangku kepentingan, peran pemangku kepentingan, dan daftar kebutuhan serta prioritas implementasinya.

3.1.4 Spesifikasi kebutuhan

Setelah mendapatkan daftar kebutuhan dan prioritas implementasi dari tahap analisa kebutuhan, maka selanjutnya dilakukan spesifikasi kebutuhan untuk merincikan kebutuhan yang akan diimplementasikan. Adapun aktivitas yang dilakukan adalah mengidentifikasi aktor yang terlibat dalam penggunaan sistem, fitur yang dimiliki sistem, kebutuhan fungsional maupun non fungsional, pemodelan diagram *usecase*, penjabaran *usecase scenario* serta pemodelan *activity diagram*.

3.1.5 Desain arsitektur sistem

Dalam mengembangkan suatu sistem, komponen struktur utama dari sistem tersebut dan hubungan antar komponennya perlu digambarkan melalui perancangan arsitektur sistem. Pada tahap ini akan digambarkan desain arsitektur Sistem Pengajuan Izin Kerja dan Praktik Tenaga Kesehatan (SINKES) yang menggunakan pola perancangan MVC.

3.1.6 Detil desain sistem

Pada tahap ini akan dilakukan perancangan sistem berdasarkan kebutuhan fungsional dan non fungsional yang telah didefinisikan. Perancangan fungsional sistem akan direpresentasikan menggunakan diagram UML seperti *sequence diagram* dan *class diagram*, sedangkan perancangan *database* menggunakan *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM). Langkah selanjutnya adalah membuat desain *user interface* (UI) atau tampilan antarmuka pengguna untuk memberikan gambaran mengenai wujud sistem yang akan dibangun. Desain *user interface* (UI) nantinya akan merepresentasikan tata letak dari tombol, fungsi dan penampilan data yang ada pada sistem. Aktivitas terakhir yang dilakukan pada tahap ini adalah perancangan algoritme pemrograman menggunakan *pseudocode* sebagai acuan pengkodean yang dilakukan pada tahap berikutnya.

3.1.7 Implementasi

Tahap implementasi sistem merupakan proses pengkodean daftar kebutuhan fungsional yang telah didefinisikan pada tahap Detil Desain Sistem. Berikut ini adalah langkah-langkah yang ada pada tahap implementasi prototype adalah sebagai berikut:

1. Implementasi desain *user interface* (UI)
Desain UI diimplementasikan pertama dengan maksud mempermudah proses implementasi kode, karena hasil implementasi tersebut dapat langsung terlihat melalui tampilan yang ada.
2. Implementasi *physical data model* (PDM)
Selanjutnya dilakukan proses pembuatan *database* berdasarkan PDM yang telah dibuat dalam tahap perancangan. *Database* ini berfungsi sebagai media penyimpanan data yang ada pada Sistem Informasi Manajemen Tenaga Kesehatan.
3. Implementasi fungsi
Langkah terakhir yang akan dilakukan adalah melakukan implementasi fungsi berdasarkan diagram *class* yang telah didefinisikan di tahap Detil Desain Sistem.

3.1.8 Pengujian sistem

Setelah sistem dikembangkan dan setiap fungsional yang didefinisikan telah dapat berjalan, maka dilakukan pengujian sistem untuk mengetahui tingkat keberhasilan sistem ketika digunakan, juga untuk menilai kesesuaiannya dengan

perancangan yang telah dibuat sebelumnya. Masing-masing *test case* (kasus uji) yang disusun berdasarkan jenis pengujian *user acceptance testing*, sedangkan *Unit Testing*, *Integration Testing* dan *System Testing* tidak akan dibahas lebih lanjut

3.1.9 Kesimpulan dan saran

Tahap akhir yang akan dilakukan dalam penelitian ini adalah pengambilan kesimpulan dari hasil penelitian yang didapat secara keseluruhan, serta pemberian saran yang ditujukan sebagai acuan penelitian selanjutnya.



BAB 4 ANALISIS KEBUTUHAN

4.1 *Business perspective*

Business perspective merupakan analisa yang dilakukan dengan mengambil sudut pandang bisnis seperti analisa terhadap proses bisnis yang sedang berjalan dan pemodelannya, analisa terhadap permasalahan yang muncul, proses bisnis yang disarankan untuk menyelesaikan permasalahan hingga analisa kebutuhan dari setiap pemangku kepentingan yang terlibat. Analisis dari *business perspective* penting dilakukan guna sebagai landasan pengembangan agar sistem yang dibuat selaras dengan kebutuhan bisnis.

4.1.1 Analisis proses bisnis *as-is*

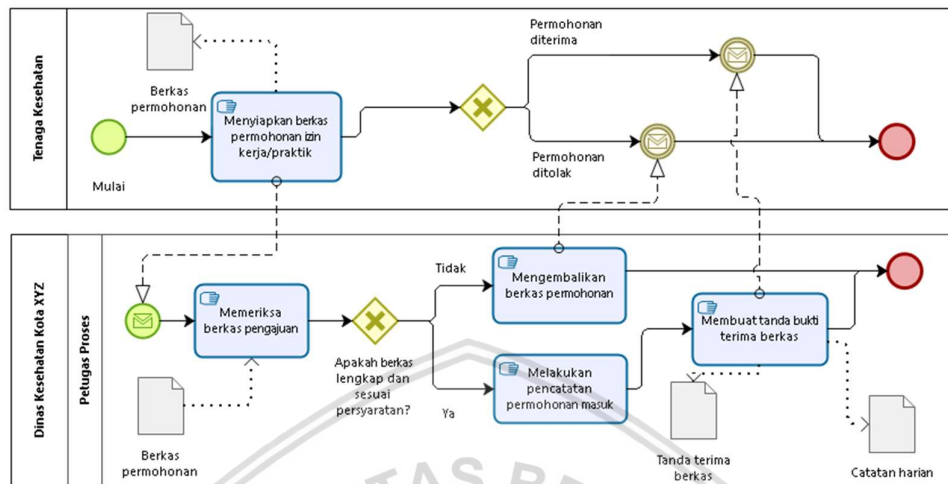
Analisis proses bisnis *as-is* dalam *business perspective* ini dilakukan untuk mengidentifikasi proses bisnis terkait pengajuan permohonan surat izin kerja/praktik oleh tenaga kesehatan yang saat ini sedang berjalan di Dinas Kesehatan Kota XYZ. Proses bisnis yang berjalan akan dimodelkan dalam diagram BPMN untuk kemudian dilakukan analisa terkait permasalahan yang ada. Dari hasil analisis permasalahan tersebut, maka selanjutnya didapatkan solusi yang akan dimodelkan dalam proses bisnis *to-be*. Berdasarkan hasil wawancara dengan pihak CV. Sarana Utama Solusindo yang mendapatkan informasi dari Petugas Proses sebagai petugas yang menangani permohonan izin kerja/praktik, diketahui bahwa terdapat 5 proses bisnis berbeda dalam menangani permohonan masing-masing tenaga kesehatan yang berjalan saat ini. Adapun 5 proses yang berbeda tersebut adalah proses bisnis permohonan izin oleh kelompok Tenaga Kesehatan jenis Apoteker, Bidan, Tenaga Teknis Kefarmasian, Perawat dan Dokter. Selain 5 proses bisnis tersebut, terdapat 1 proses bisnis pengajuan berkas yang sama untuk semua tenaga kesehatan yang telah disebutkan sebelumnya.

4.1.1.1 Proses bisnis *as-is* pengajuan permohonan

Tenaga Kesehatan jenis Apoteker, Bidan, Dokter, Perawat dan Tenaga Teknis Kefarmasian memiliki proses bisnis pengajuan berkas permohonan sama yang melibatkan tenaga kesehatan dan Petugas Proses yang berperan sebagai petugas administrasi. Proses bisnis tersebut digambarkan dalam Gambar 4.1 dan adapun penjelasan dari proses yang berjalan adalah sebagai berikut.

1. Pemohon datang ke Kantor Dinas Kesehatan Kota XYZ membawa berkas persyaratan yang dibutuhkan dan menyerahkannya kepada Petugas Proses.
2. Petugas Proses memeriksa kelengkapan berkas yang diajukan pemohon.
 - a. Jika berkas dinyatakan tidak lengkap, maka berkas akan dikembalikan kepada pemohon untuk dilengkapi.
 - b. Jika berkas yang diajukan telah dinyatakan lengkap dan sesuai persyaratan, maka Petugas Proses akan menuliskan data pemohon

di buku catatan harian, kemudian membuat tanda bukti penerimaan berkas yang akan diberikan kepada pemohon.



Gambar 4.1 Proses bisnis *as-is* pengajuan permohonan

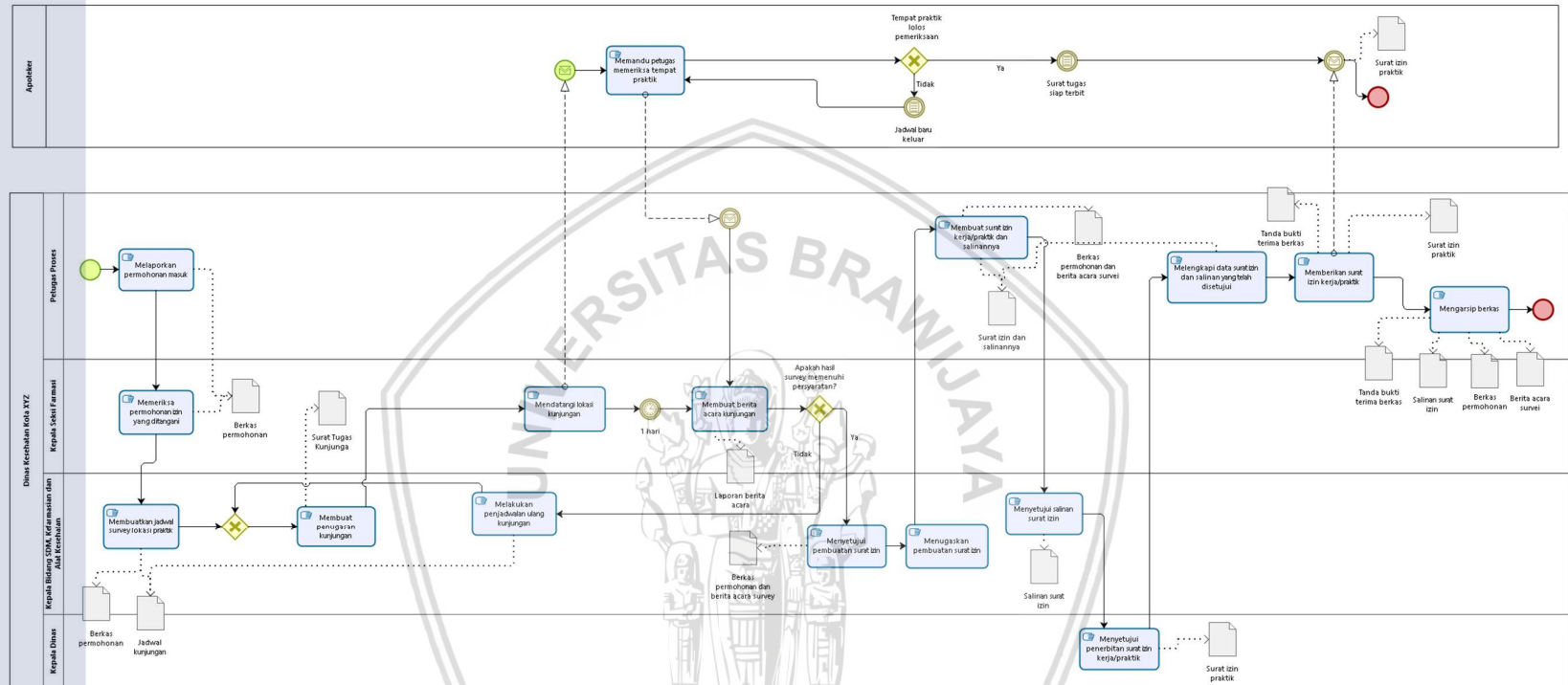
4.1.1.2 Proses bisnis *as-is* penanganan permohonan izin praktik Apoteker

Pada proses bisnis penanganan permohonan izin praktik untuk tenaga kesehatan jenis Apoteker, terdapat 4 pegawai dari Dinas Kesehatan Kota XYZ yang terlibat yaitu Petugas Proses; Kepala Seksi Farmasi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas. Proses bisnis ditunjukkan pada Gambar 4.2 dan penjelasannya sebagai berikut.

1. Setelah menerima berkas permohonan dari tenaga kesehatan jenis Apoteker, Petugas proses melaporkan kepada Kepala Seksi Kefarmasian terkait permohonan izin praktik yang masuk.
2. Kepala Seksi Kefarmasian melaporkan kepada Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan terkait permohonan yang masuk. Selanjutnya Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan membuat jadwal survei lokasi praktik dan menugaskan Kepala Seksi Kefarmasian untuk melakukan survei lokasi tersebut.
3. Kepala Seksi Kefarmasi mengunjungi tempat praktik Apoteker untuk memeriksa kelayakan lokasi. Setelah survei berlangsung, Kepala Seksi Kefarmasian membuat berita acara yang berisi penilaian kelayakan untuk dilaporkan kepada Kepala Bidang Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan.
4. Jika dari hasil penilaian dinyatakan layak, maka Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan akan menugaskan Petugas Proses untuk membuat Surat Izin Praktik Apoteker. Namun jika dinyatakan tidak layak, maka Apoteker diberikan waktu untuk memperbaiki lokasi praktik dan mendapatkan jadwal survei lagi.

5. Petugas proses mencetak formulir surat izin dan salinannya untuk kemudian diisi secara manual. Setelah surat izin dan salinannya terisi, maka Petugas Proses akan membawa salinannya untuk diparaf oleh Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan, serta membawa surat izin yang asli ke Kepala Dinas Kesehatan untuk ditandatangani.
6. Setelah surat izin siap diterbitkan, Petugas Proses mengarsip berkas pemohon dan salinan surat izin.
7. Pemohon datang lagi ke Kantor Dinas Kesehatan Kota XYZ untuk mengambil Surat Izin Praktik yang telah jadi dengan membawa tanda bukti terima berkas yang didapat sebelumnya.





Gambar 4.2 Proses bisnis *as-is* penanganan permohonan izin praktik Apoteker

4.1.1.3 Proses bisnis *as-is* penanganan permohonan izin praktik Bidan

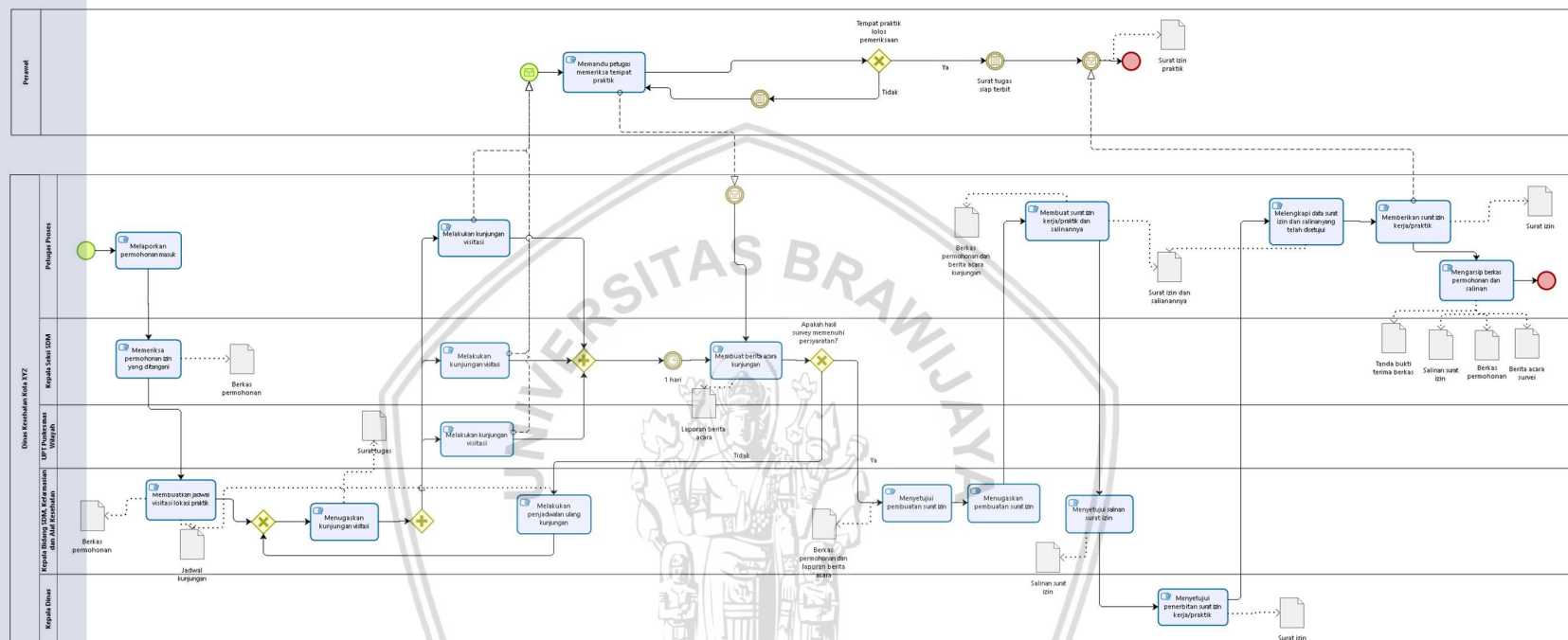
Pada proses bisnis penanganan permohonan izin praktik untuk tenaga kesehatan jenis Bidan, terdapat 5 pegawai dari Dinas Kesehatan Kota XYZ yang terlibat yaitu Petugas Proses; Kepala Seksi SDM; Seksi Kesehatan Keluarga dan Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas. Proses bisnis ditunjukkan pada Gambar 4.3 dan penjelasannya sebagai berikut.

1. Setelah menerima berkas permohonan dari tenaga kesehatan jenis Bidan, Petugas proses melaporkan kepada Kepala Seksi Kefarmasian terkait permohonan izin praktik yang masuk.
2. Kepala Seksi SDM melaporkan kepada Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan terkait permohonan yang masuk. Selanjutnya Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan membuat jadwal survei lokasi praktik dan menugaskan Petugas Proses, Kepala Seksi SDM dan Seksi Kesehatan Keluarga dan Gizi untuk melakukan survei lokasi tersebut.
3. Petugas yang ditugaskan melakukan kunjungan mengunjungi tempat praktik Bidan untuk memeriksa kelayakan lokasi. Setelah survei berlangsung, Kepala Seksi SDM membuat berita acara yang berisi penilaian kelayakan untuk dilaporkan kepada Kepala Bidang Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan.
4. Jika dari hasil penilaian dinyatakan layak, maka Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan akan menugaskan Petugas Proses untuk membuat Surat Izin Praktik Bidan. Namun jika dinyatakan tidak layak, maka Bidan diberikan waktu untuk memperbaiki lokasi praktik dan mendapatkan jadwal survei lagi.
5. Petugas proses mencetak formulir surat izin dan salinannya untuk kemudian diisi secara manual. Setelah surat izin dan salinannya terisi, maka Petugas Proses akan membawa salinannya untuk diparaf oleh Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan, serta membawa surat izin yang asli ke Kepala Dinas Kesehatan untuk ditandatangani.
6. Setelah surat izin siap diterbitkan, Petugas Proses mengarsip berkas pemohon dan salinan surat izin.
7. Pemohon datang lagi ke Kantor Dinas Kesehatan Kota XYZ untuk mengambil Surat Izin Praktik yang telah jadi dengan membawa tanda bukti terima berkas yang didapat sebelumnya.

4.1.1.4 Proses bisnis *as-is* penanganan permohonan izin kerja Perawat

Pada proses bisnis penanganan permohonan izin kerja untuk tenaga kesehatan jenis Perawat, terdapat 5 pegawai dari Dinas Kesehatan Kota XYZ yang terlibat yaitu Petugas Proses; Kepala Seksi SDM; UPT Puskesmas Wilayah; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas. Proses bisnis ditunjukkan pada Gambar 4.4 dan penjelasannya sebagai berikut.

1. Setelah menerima berkas permohonan dari tenaga kesehatan jenis Perawat, Petugas proses melaporkan kepada Kepala Seksi SDM terkait permohonan izin praktik Bidan yang masuk.
2. Kepala Seksi SDM melaporkan kepada Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan terkait permohonan yang masuk. Selanjutnya Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan membuat jadwal survei lokasi praktik dan menugaskan Petugas Proses, Kepala Seksi SDM dan UPT Puskesmas Wilayah untuk melakukan survei lokasi tersebut.
3. Petugas mengunjungi tempat praktik Perawat untuk memeriksa kelayakan lokasi. Setelah survei berlangsung, Kepala Seksi SDM membuat berita acara yang berisi penilaian kelayakan untuk dilaporkan kepada Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan.
4. Jika dari hasil penilaian dinyatakan layak, maka Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan akan menugaskan Petugas Proses untuk membuat Surat Izin Praktik Perawat. Namun jika dinyatakan tidak layak, maka Perawat diberikan waktu untuk memperbaiki lokasi praktik dan mendapatkan jadwal survei lagi.
5. Petugas proses mencetak formulir surat izin dan salinannya untuk kemudian diisi secara manual. Setelah surat izin dan salinannya terisi, maka Petugas Proses akan membawa salinannya untuk diparaf oleh Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan, serta membawa surat izin yang asli ke Kepala Dinas Kesehatan untuk ditandatangani.
6. Setelah surat izin siap diterbitkan, Petugas Proses mengarsip berkas pemohon dan salinan surat izin.
7. Pemohon datang lagi ke Kantor Dinas Kesehatan Kota XYZ untuk mengambil Surat Izin Praktik yang telah jadi dengan membawa tanda bukti terima berkas yang didapat sebelumnya.

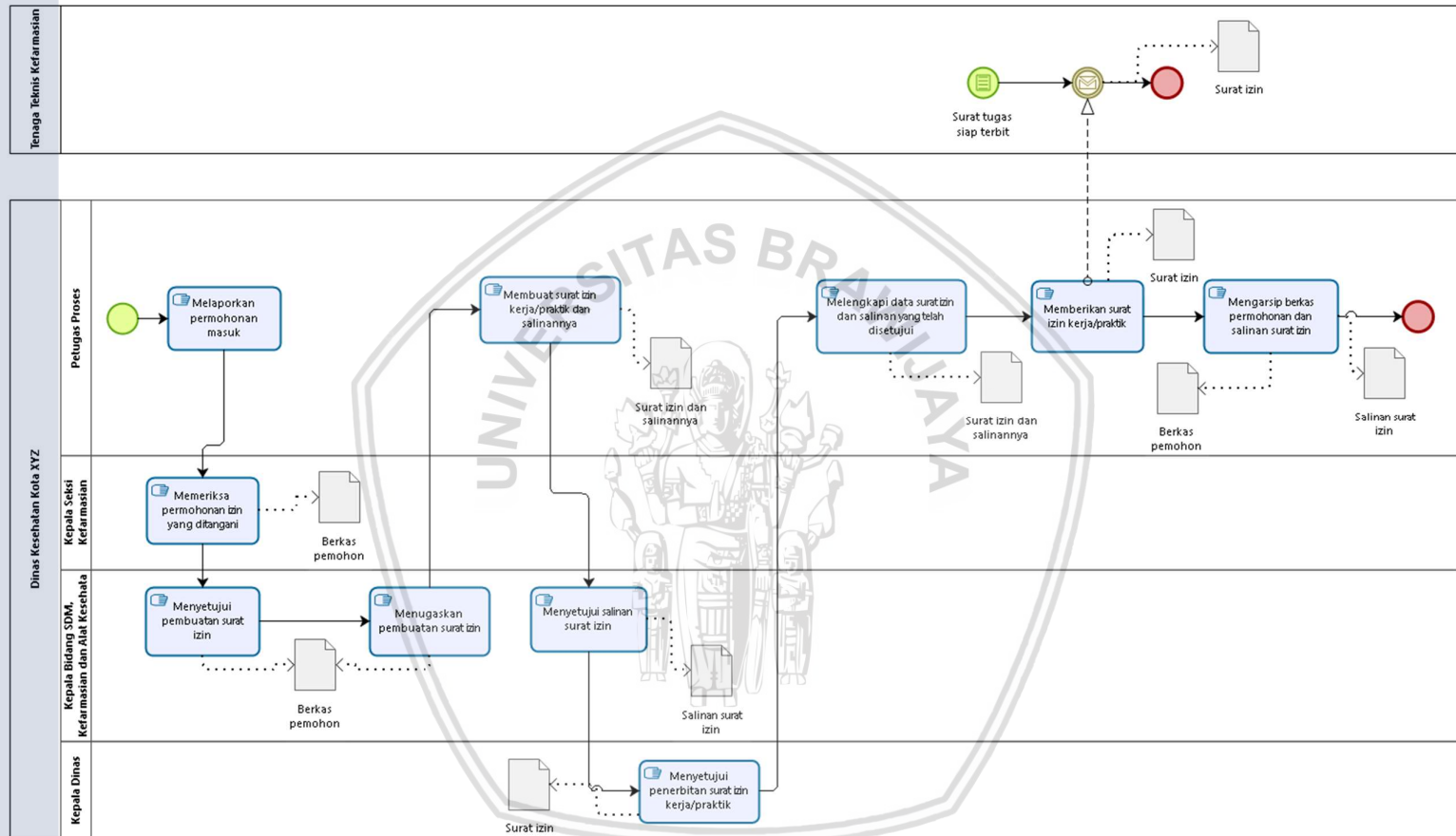


Gambar 4.4 Proses bisnis *as-is* penanganan permohonan izin praktik Perawat

4.1.1.5 Proses bisnis *as-is* penanganan permohonan izin kerja Tenaga Teknis Kefarmasian

Pada proses bisnis penanganan permohonan izin kerja untuk tenaga kesehatan jenis Tenaga Teknis Kesehatan, terdapat 4 pegawai dari Dinas Kesehatan Kota XYZ yang terlibat yaitu Petugas Proses; Kepala Seksi Kefarmasian; Seksi Kesehatan Keluarga dan Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas. Proses bisnis ditunjukkan pada Gambar 4.5 dan penjelasannya sebagai berikut.

1. Setelah menerima berkas permohonan dari tenaga kesehatan jenis Tenaga Teknis Kefarmasian, Petugas proses melaporkan kepada Kepala Seksi Kefarmasian terkait permohonan izin yang masuk.
2. Kepala Seksi SDM melaporkan kepada Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan terkait permohonan yang masuk.
3. Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan menugaskan Petugas Proses untuk membuatkan Surat Izin Kerja Tenaga Teknis Kefarmasian.
4. Petugas proses mencetak formulir surat izin dan salinannya untuk kemudian diisi secara manual. Setelah surat izin dan salinannya terisi, maka Petugas Proses akan membawa salinannya untuk diparaf oleh Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan, serta membawa surat izin yang asli ke Kepala Dinas Kesehatan untuk ditandatangani.
5. Setelah surat izin siap diterbitkan, Petugas Proses mengarsip berkas pemohon dan salinan surat izin.
6. Pemohon datang lagi ke Kantor Dinas Kesehatan Kota XYZ untuk mengambil Surat Izin Praktik yang telah jadi dengan membawa tanda bukti terima berkas yang didapat sebelumnya.

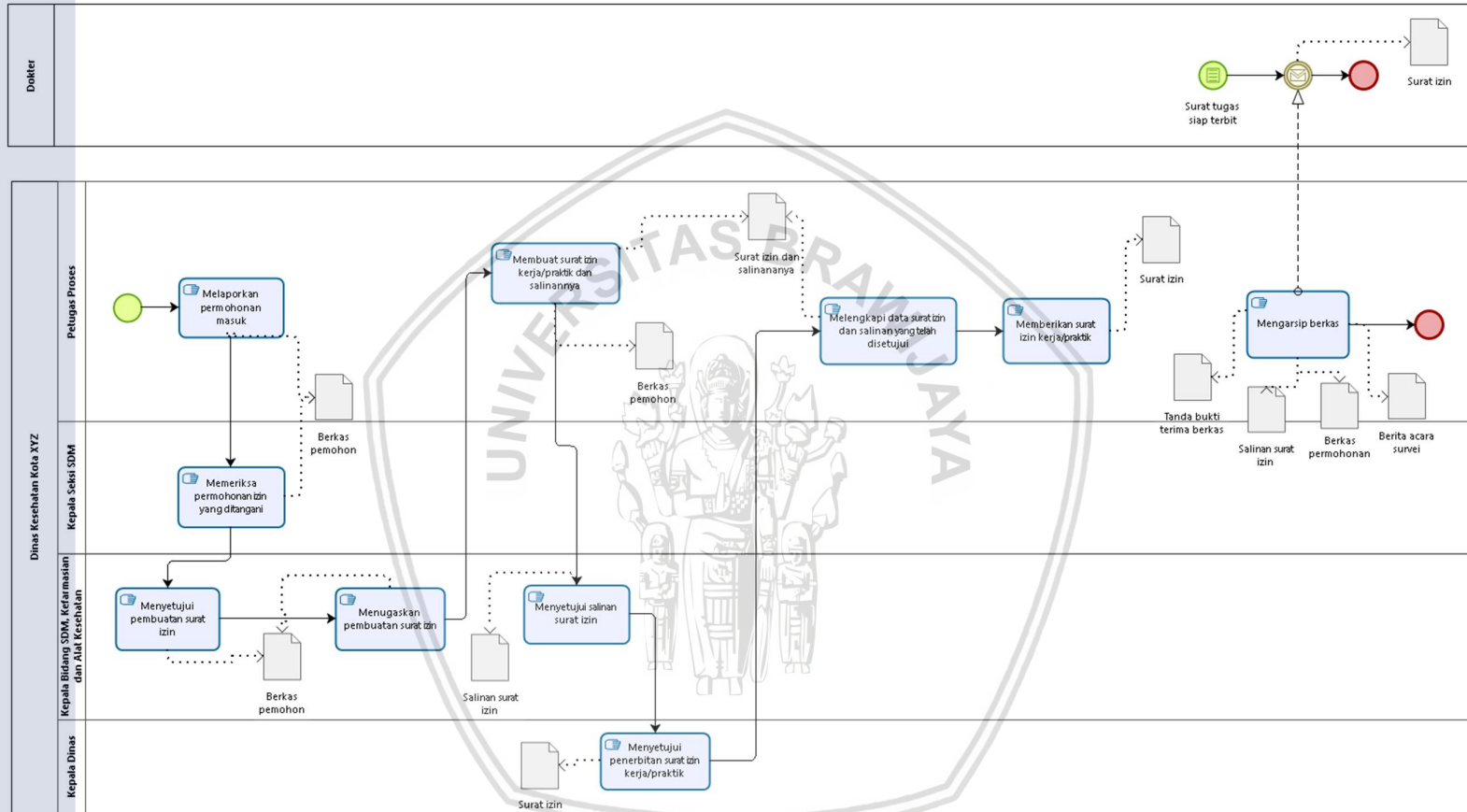


Gambar 4.5 Proses bisnis *as-Is* penanganan permohonan izin kerja Tenaga Teknis Kefarmasian

4.1.1.6 Proses bisnis *as-is* penanganan permohonan izin praktik Dokter

Pada proses bisnis penanganan permohonan izin praktik untuk tenaga kesehatan jenis Dokter, terdapat 4 pegawai dari Dinas Kesehatan Kota XYZ yang terlibat yaitu Petugas Proses; Kepala Seksi SDM; Seksi Kesehatan Keluarga dan Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas. Proses bisnis ditunjukkan pada Gambar 4.6 dan penjelasannya sebagai berikut.

1. Setelah menerima berkas permohonan dari tenaga kesehatan jenis Dokter, Petugas proses melaporkan kepada Kepala Seksi SDM terkait permohonan izin kerja/praktik yang masuk.
2. Kepala Seksi SDM melaporkan kepada Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan terkait permohonan yang masuk.
3. Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan menugaskan Petugas Proses untuk membuat Surat Izin Kerja/Praktik tenaga kesehatan.
4. Petugas proses mencetak formulir surat izin dan salinannya untuk kemudian diisi secara manual. Setelah surat izin dan salinannya terisi, maka Petugas Proses akan membawa salinannya untuk diparaf oleh Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan, serta membawa surat izin yang asli ke Kepala Dinas Kesehatan untuk ditandatangani.
5. Setelah surat izin siap diterbitkan, Petugas Proses mengarsip berkas pemohon dan salinan surat izin.
6. Pemohon datang lagi ke Kantor Dinas Kesehatan Kota XYZ untuk mengambil Surat Izin Praktik yang telah jadi dengan membawa tanda bukti terima berkas yang didapat sebelumnya.



Gambar 4.6 Proses bisnis *as-is* penanganan permohonan izin praktik Dokter

4.1.2 Analisis permasalahan

Pada bagian analisis permasalahan berikut dijelaskan mengenai permasalahan yang dihadapi saat ini pada proses pengajuan dan penanganan permohonan izin kerja/praktik dari tenaga kesehatan yang bekerja dibawah area kerja Dinas Kesehatan Kota XYZ. Permasalahan yang dijelaskan didapatkan dari hasil wawancara dengan CV. Sarana Utama Solusindo yang ditunjukkan pada Lampiran A. Tabel 4.1 menjelaskan tentang analisis permasalahan yang dihadapi.

Tabel 4.1 Analisis Permasalahan

Masalah	Pemilik
1. Tenaga kesehatan harus mendatangi Kantor Dinkes berkali-kali jika terdapat berkas persyaratan yang kurang.	Tenaga Kesehatan
2. Tenaga kesehatan hanya diberi waktu estimasi surat izin kerja/praktik siap untuk diambil, sehingga tenaga kesehatan perlu memeriksa ke Kantor Dinkes secara langsung atau melalui telepon.	Tenaga Kesehatan
3. Petugas proses yang menuliskan data pemohon secara manual ke buku harian, tanda terima berkas, maupun surat izin kerja/praktik, sehingga rawan terjadi kesalahan penulisan.	Petugas Proses
4. Petugas proses harus mendatangi ruangan masing-masing kepala bagian/seksi dengan membawa berkas pemohon, sehingga petugas proses tidak selalu berada di tempat dan berkas rawan hilang karena terselip.	Petugas Proses
5. Berkas yang harus diperiksa berupa berkas cetak, sehingga petugas yang sedang tidak berada di Kantor Dinkes tidak dapat memeriksa atau memberi persetujuan.	Kepala Seksi dan Kepala Bidang
6. Pihak Dinkes yang hanya memiliki salinan cetak berkas permohonan, sehingga tidak memiliki arsip cadangan.	Petugas Proses
Mempengaruhi	
Kinerja pegawai Dinas Kesehatan Kota XYZ dalam bertugas melayani permintaan surat izin kerja/praktik.	
Dampak	
1. Beberapa tenaga kesehatan tidak segera kembali ke Kantor Dinkes untuk mengurus perizinan ketika ada berkas yang kurang dari persyaratan. 2. Petugas di Kantor Dinkes mendapatkan komplain karena dianggap lamban dalam melayani masyarakat. 3. Data yang kurang akurat akibat terjadinya kesalahan penulisan. 4. Tenaga kesehatan yang akan mengajukan permohonan tidak segera terlayani. 5. Mengulur waktu penanganan permohonan yang diajukan oleh tenaga kesehatan. 6. Jika terdapat arsip yang hilang/terselip, maka pihak Dinkes akan kesulitan saat ingin melakukan pendataan.	
Solusi	

Tabel 4.1 Analisis Permasalahan

1.	Menyediakan sistem yang dapat digunakan oleh tenaga kesehatan untuk mengajukan permohonan tanpa harus datang langsung ke Kantor Dinkes.
2.	Menyediakan sistem yang dapat digunakan tenaga kesehatan untuk memeriksa tahapan penanganan surat izin yang diajukan ketika sedang berada dimanapun dan kapanpun.
3.	Menyediakan sistem yang dapat membantu petugas proses membuat dokumen untuk kebutuhan penanganan permohonan yang diajukan.
4.	Menyediakan sistem yang memungkinkan petugas Dinkes untuk melakukan pemeriksaan/persetujuan berkas tanpa petugas proses perlu ke ruangan masing-masing kepala bidang atau seksi dengan membawa banyak berkas.
5.	Menyediakan sistem yang dapat memfasilitasi kepala seksi dan bidang untuk memeriksa berkas kapanpun dan dimanapun.
6.	Menyediakan sistem yang dapat menyimpan data pemohon untuk memudahkan pemeriksaan sewaktu-waktu.

Berdasarkan hasil observasi dokumen pendukung terkait proses bisnis yang saat ini berjalan, berikut adalah analisis waktu yang diperlukan untuk mengajukan permohonan hingga surat izin siap terbit jika pengajuan dan proses penanganan berjalan dengan sebagaimana mestinya.

Tabel 4.2 Analisis waktu proses yang dibutuhkan

No.	Identifikasi Masalah	Waktu
1.	Penyerahan berkas permohonan ke Petugas Proses di Kantor Dinkes.	15 menit
2.	Memeriksa kelengkapan berkas.	20 menit
3.	Pembuatan tanda bukti terima berkas dan penyampaian tahapan penanganan yang akan dilalui.	10 menit
4.	Persetujuan tahap berikutnya oleh Kepala Seksi.	15 menit
5.	Penjadwalan kunjungan survei/visitasi.	30 menit
6.	Kunjungan survei/visitasi.	1 jam
7.	Pembuatan laporan berita acara.	1 jam
8.	Penentuan lolos atau tidaknya kunjungan yang telah dilakukan.	15 menit
9.	Penandatanganan Salinan surat izin oleh Kepala Bidang.	5 menit
10.	Penandatanganan surat izin oleh Kepala Dinas.	5 menit
11.	Pengarsipan berkas pemohon.	10 menit

4.1.3 Identifikasi proses bisnis *to-be*

Dari hasil analisis permasalahan yang ada pada proses bisnis *as-is* dan solusi yang diusulkan, selanjutnya dilakukan identifikasi proses bisnis *to-be* untuk memberikan penjelasan dan gambaran kepada *stakeholder* organisasi terhadap usulan-usulan yang diberikan guna mengetahui apakah telah sesuai dengan yang diharapkan.

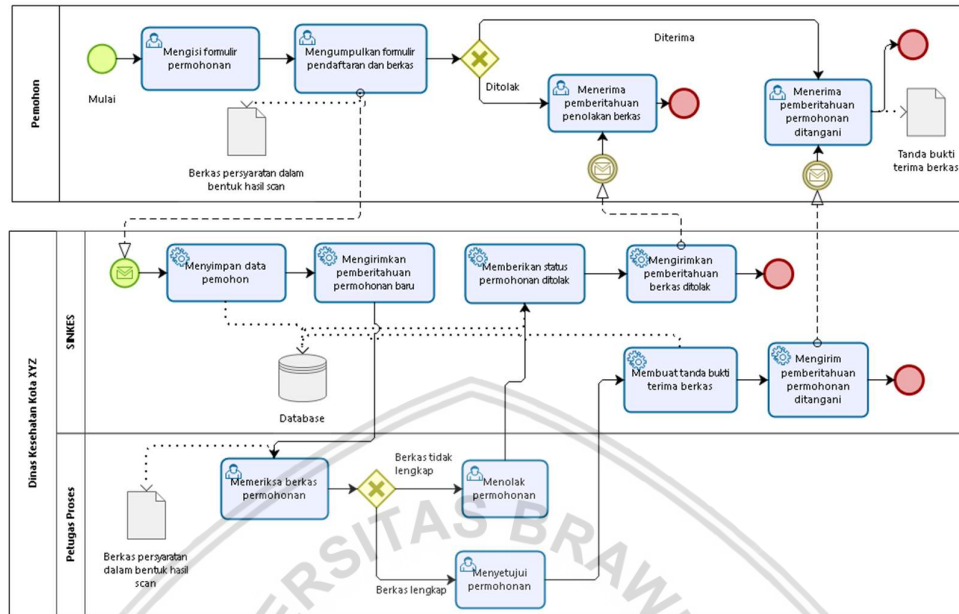
4.1.3.1 Proses bisnis pengajuan permohonan *to-be*

Pemodelan proses bisnis *to-be* merupakan gambaran dari proses bisnis usulan yang dibuat berdasarkan solusi yang ditawarkan untuk menyelesaikan permasalahan pada Tabel 4.2. Proses bisnis ditunjukkan pada Gambar 4.7 dan penjelasannya sebagai berikut.

1. Tenaga kesehatan mengisi formulir pengajuan dengan data diri dan berkas persyaratan dalam bentuk hasil *scan*.
2. Tenaga kesehatan mengirimkan formulir dan berkas persyaratan.
3. Petugas Proses Dinkes menerima pengajuan, kemudian melakukan pemeriksaan data diri dan berkas persyaratan yang diajukan.
4. Jika persyaratan terpenuhi dan disetujui oleh Petugas Proses, maka sistem akan membuat tanda bukti terima berkas dan dikirimkan kepada Tenaga Kesehatan yang bersangkutan.
5. Jika berkas tidak memenuhi persyaratan, maka permohonan akan ditolak dan tenaga kesehatan mendapatkan pemberitahuan.

Tabel 4.3 Perubahan aktivitas proses bisnis pengajuan permohonan

Unit	Proses Bisnis As-Is	Proses Bisnis To-Be	Keterangan
Tenaga Kesehatan	Memberikan berkas permohonan kepada Petugas Proses.	Sistem menyediakan formulir untuk mengajukan permohonan.	Diubah.
Petugas Proses	Memeriksa kertas berkas persyaratan.	Sistem menyediakan halaman untuk memeriksa berkas yang dilampirkan.	Diubah.
	Mencatatkan permohonan masuk pada buku catatan harian.	Sistem secara otomatis menyimpan <i>record</i> penerimaan permohonan yang diajukan.	Diubah.
	Membuat tanda terima berkas pada blangko yang tersedia.	Sistem secara otomatis mengirimkan tanda bukti terima berkas kepada Tenaga Kesehatan saat pengajuan disetujui.	Diubah.



Gambar 4.7 Pemodelan Proses Bisnis Pengajuan Permohonan To-Be

4.1.3.2 Proses bisnis penanganan permohonan izin praktik Apoteker *to-be*

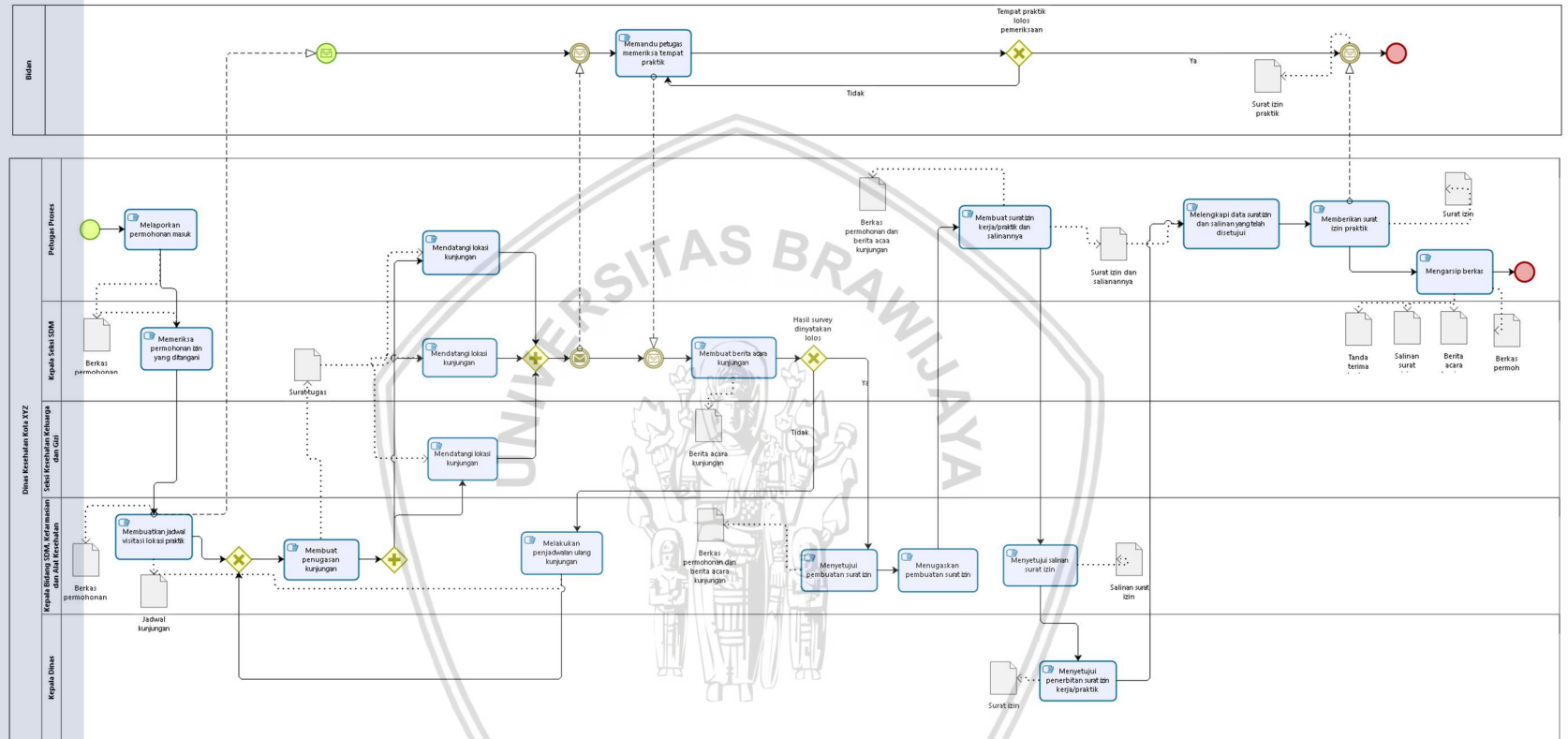
Proses bisnis penanganan permohonan izin praktik *to-be* untuk tenaga kesehatan jenis Apoteker tetap melibatkan 4 pegawai dari Dinas Kesehatan Kota XYZ yaitu Petugas Proses; Kepala Seksi Kefarmasian; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas. Pada proses bisnis *to-be* yang menjadi usulan, proses pengajuan dan penanganan permohonan digambarkan menjadi 2 gambar terpisah, gambar penanganan permohonan ditunjukkan pada Gambar 4.8 dan penjelasannya sebagai berikut.

1. Setelah Petugas Proses menyetujui berkas permohonan yang diajukan, maka sistem akan mengirimkan pemberitahuan dan data terkait kepada Kepala Seksi Kefarmasian.
2. Kepala Seksi Kefarmasian memberikan persetujuan untuk selanjutnya sistem mengirimkan pemberitahuan terkait permohonan ke Kepala Bidang SDM, Kefarmasian & Alat Kesehatan untuk selanjutnya dibuatkan jadwal kunjungan ke tempat praktik.
3. Sistem mengirimkan pemberitahuan kepada pemohon waktu pelaksanaan kunjungan oleh pihak Dinkes dan penugasan kunjungan kepada Kepala Seksi Kefarmasian.
4. Kepala Seksi Kefarmasian melakukan kunjungan ke tempat praktik, kemudian membuat berita acara kunjungan.
5. Jika hasil kunjungan dinyatakan memenuhi syarat, maka Kepala Bidang SDM, Kefarmasian & Alat Kesehatan akan menugaskan Petugas Proses untuk mencetak surat izin praktik.
6. Jika tidak, maka Kepala Bidang SDM, Kefarmasian & Alat Kesehatan akan melakukan penjadwalan ulang kunjungan kepada pemohon.
7. Petugas Proses mencetak surat izin praktik dan salinannya, kemudian meminta persetujuan kepada Kepala Bidang SDM, Kefarmasian & Alat Kesehatan dan Kepala Dinas.
8. Setelah berkas surat disetujui, salinan surat izin akan diarsip kemudian Petugas Proses merubah status penanganan untuk memberitahukan bahwa surat izin telah siap terbit ke Pemohon.
9. Pemohon datang ke Kantor Dinkes membawa hasil cetak tanda bukti penerimaan berkas dan berkas persyaratan untuk mendapatkan surat izin praktik yang telah siap terbit.
10. Petugas proses menambahkan tanda bukti terima berkas dari pemohon ke arsip.

4.1.3.3 Proses bisnis penanganan permohonan izin praktik Bidan *to-be*

Proses bisnis pengajuan permohonan izin praktik *to-be* untuk tenaga kesehatan jenis Bidan tetap melibatkan 5 pegawai dari Dinas Kesehatan Kota XYZ yaitu Petugas Proses; Kepala Seksi SDM; Seksi Kesehatan Keluarga & Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas. Pada proses bisnis *to-be* yang menjadi usulan, proses pengajuan dan penanganan permohonan digambarkan menjadi 2 gambar terpisah, gambar penanganan permohonan ditunjukkan pada Gambar 4.9 dan penjelasannya sebagai berikut.

1. Setelah Petugas Proses menyetujui berkas permohonan yang diajukan, maka sistem akan mengirimkan pemberitahuan dan data terkait kepada Kepala Seksi SDM.
2. Kepala Seksi SDM memberikan persetujuan untuk selanjutnya sistem mengirimkan pemberitahuan terkait permohonan ke Kepala Bidang SDM, Kefarmasian & Alat Kesehatan untuk selanjutnya dibuatkan jadwal kunjungan ke tempat praktik.
3. Sistem mengirimkan pemberitahuan kepada pemohon waktu pelaksanaan kunjungan oleh pihak Dinkes dan penugasan kunjungan kepada Kepala Seksi SDM, Seksi Kesehatan Keluarga & Gizi, serta Petugas Proses.
4. Kepala Seksi SDM, Seksi Kesehatan Keluarga & Gizi, serta Petugas Proses melakukan kunjungan ke tempat praktik, kemudian Kepala Seksi SDM membuat berita acara kunjungan.
5. Jika hasil kunjungan dinyatakan memenuhi syarat, maka Kepala Bidang SDM, Kefarmasian & Alat Kesehatan akan menugaskan Petugas Proses untuk mencetak surat izin praktik.
6. Jika tidak, maka Kepala Bidang SDM, Kefarmasian & Alat Kesehatan akan melakukan penjadwalan ulang kunjungan kepada pemohon.
7. Petugas Proses mencetak surat izin praktik dan salinannya, kemudian meminta persetujuan kepada Kepala Bidang SDM, Kefarmasian & Alat Kesehatan dan Kepala Dinas.
8. Setelah berkas surat disetujui, salinan surat izin akan diarsip kemudian Petugas Proses merubah status penanganan untuk memberitahukan bahwa surat izin telah siap terbit ke Pemohon.
9. Pemohon datang ke Kantor Dinkes membawa hasil cetak tanda bukti penerimaan berkas dan berkas persyaratan untuk mendapatkan surat izin praktik yang telah siap terbit.
10. Petugas proses menambahkan tanda bukti terima berkas dari pemohon ke arsip.



Gambar 4.9 Pemodelan proses bisnis pengajuan izin praktik Bidan *to-be*

4.1.3.4 Proses bisnis penanganan permohonan izin praktik Perawat *to-be*

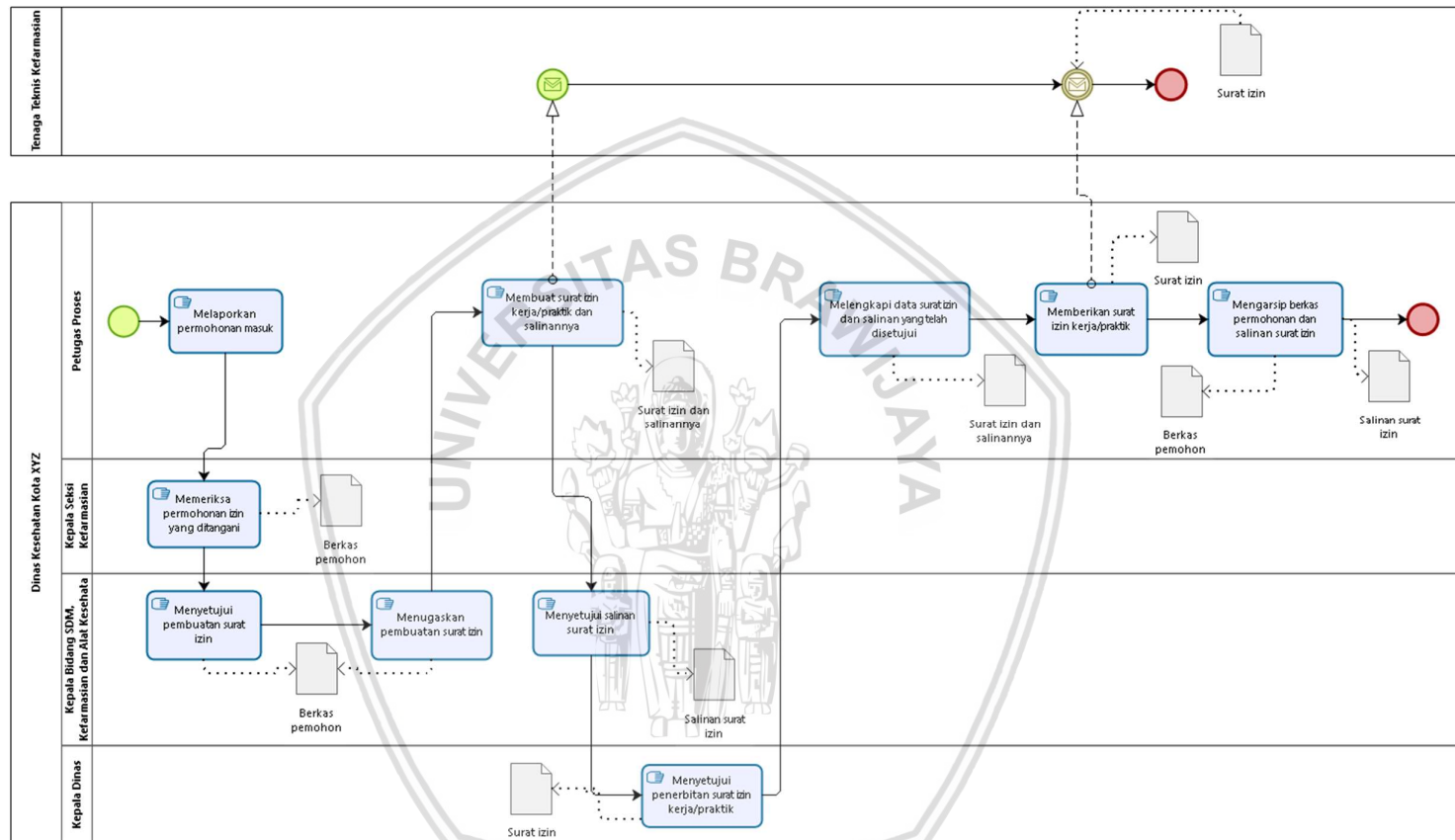
Proses bisnis pengajuan permohonan izin praktik *to-be* untuk tenaga kesehatan jenis Perawat melibatkan 4 pegawai dari Dinas Kesehatan Kota XYZ yaitu Petugas Proses; Kepala Seksi SDM; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas serta 1 perwakilan dari UPT Puskesmas Wilayah. Pada proses bisnis *to-be* yang menjadi usulan, proses pengajuan dan penanganan permohonan digambarkan menjadi 2 gambar terpisah, gambar penanganan permohonan ditunjukkan pada Gambar 4.10 dan penjelasannya sebagai berikut.

1. Setelah Petugas Proses menyetujui berkas permohonan yang diajukan, maka sistem akan mengirimkan pemberitahuan dan data terkait kepada Kepala Seksi SDM.
2. Kepala Seksi SDM memberikan persetujuan untuk selanjutnya sistem mengirimkan pemberitahuan terkait permohonan ke Kepala Bidang SDM, Kefarmasian & Alat Kesehatan untuk selanjutnya dibuatkan jadwal kunjungan ke tempat praktik.
3. Sistem mengirimkan pemberitahuan kepada pemohon waktu pelaksanaan kunjungan oleh pihak Dinkes dan penugasan kunjungan kepada Kepala Seksi SDM, UPT Puskesmas Wilayah, serta Petugas Proses.
4. Kepala Seksi SDM, UPT Puskesmas Wilayah, serta Petugas Proses melakukan kunjungan ke tempat praktik, kemudian Kepala Seksi SDM membuat berita acara kunjungan.
5. Jika hasil kunjungan dinyatakan memenuhi syarat, maka Kepala Bidang SDM, Kefarmasian & Alat Kesehatan akan menugaskan Petugas Proses untuk mencetak surat izin praktik.
6. Jika tidak, maka Kepala Bidang SDM, Kefarmasian & Alat Kesehatan akan melakukan penjadwalan ulang kunjungan kepada pemohon.
7. Petugas Proses mencetak surat izin praktik dan salinannya, kemudian meminta persetujuan kepada Kepala Bidang SDM, Kefarmasian & Alat Kesehatan dan Kepala Dinas.
8. Setelah berkas surat disetujui, salinan surat izin akan diarsip kemudian Petugas Proses merubah status penanganan untuk memberitahukan bahwa surat izin telah siap terbit ke Pemohon.
9. Pemohon datang ke Kantor Dinkes membawa hasil cetak tanda bukti penerimaan berkas dan berkas persyaratan untuk mendapatkan surat izin praktik yang telah siap terbit.
10. Petugas proses menambahkan tanda bukti terima berkas dari pemohon ke arsip.

4.1.3.5 Proses bisnis penanganan permohonan izin kerja Tenaga Teknis Kefarmasian *to-be*

Pada proses bisnis pengajuan izinkerja untuk Tenaga Teknis Kefarmasian *to-be* yang menjadi usulan, masih melibatkan 4 pegawai dari sisi Dinas Kesehatan yaitu Petugas Proses; Kepala Seksi Kefarmasian; Kepala Bidang SDM, Kefarmasian & Alat Kesehatan; serta Kepala Dinas. Proses bisnis usulan untuk Tenaga Teknis Kefarmasian ini juga dimodelkan menjadi 2 gambar terpisah yaitu proses pengajuan dan penanganan. Proses bisnis penanganan ditunjukkan pada Gambar 4.11 dan penjelasannya sebagai berikut.

1. Setelah Petugas Proses menyetujui berkas permohonan yang diajukan, maka sistem akan mengirimkan pemberitahuan dan data terkait kepada Kepala Seksi Kefarmasian.
2. Kepala Seksi Kefarmasian melaporkan kepada Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan terkait permohonan yang masuk.
3. Setelah melihat berkas yang telah memenuhi persyaratan, Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan menugaskan Petugas Proses untuk membuat Surat Izin Kerja/Praktik tenaga kesehatan.
4. Petugas Proses mencetak surat izin praktik dan salinannya, kemudian meminta persetujuan kepada Kepala Bidang SDM, Kefarmasian & Alat Kesehatan dan Kepala Dinas.
5. Setelah berkas surat disetujui, salinan surat izin akan diarsip kemudian Petugas Proses merubah status penanganan untuk memberitahukan bahwa surat izin telah siap terbit ke Pemohon.
6. Pemohon datang ke Kantor Dinkes membawa hasil cetak tanda bukti penerimaan berkas dan berkas persyaratan untuk mendapatkan surat izin praktik yang telah siap terbit.
7. Petugas proses menambahkan tanda bukti terima berkas dari pemohon ke arsip.

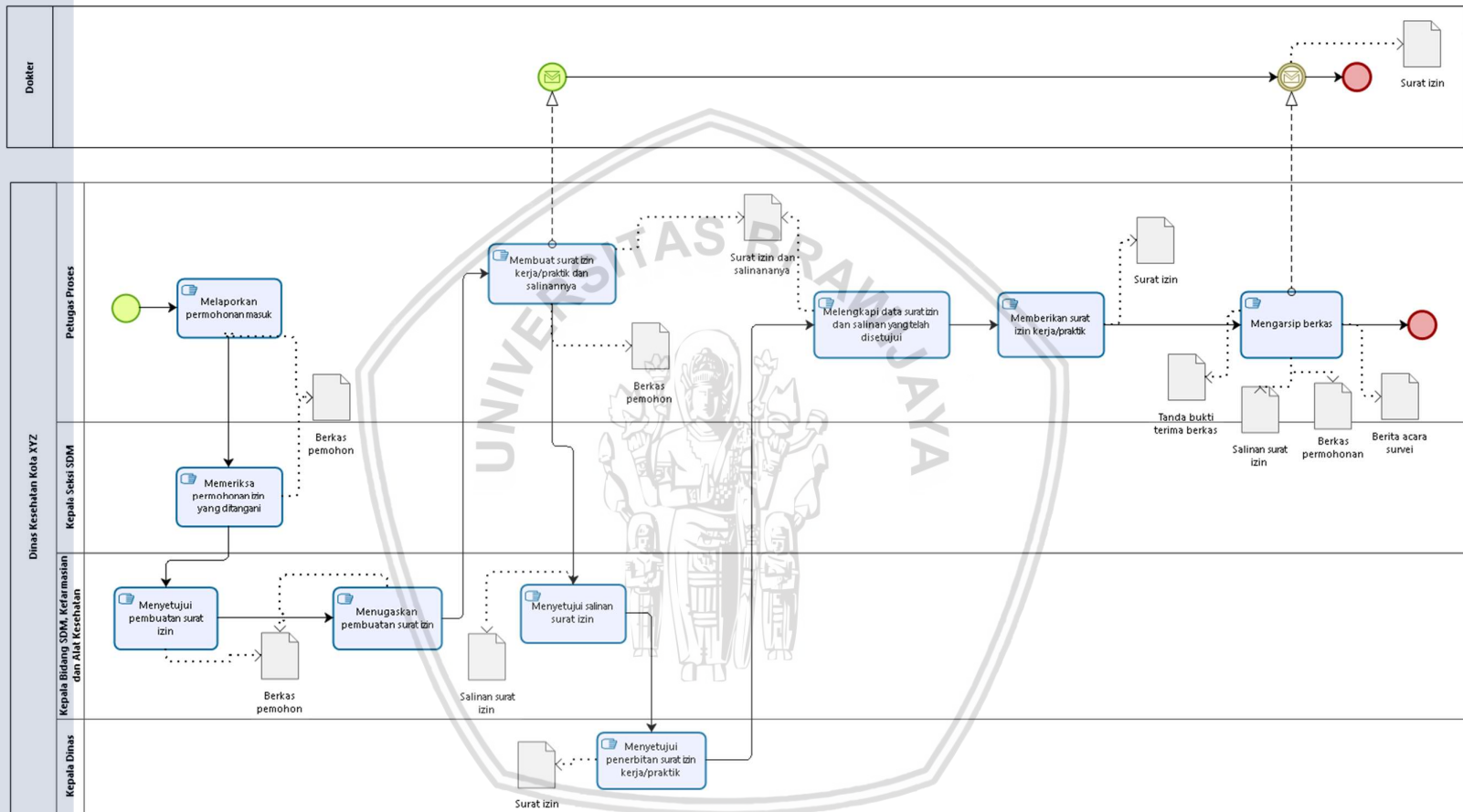


Gambar 4.11 Proses bisnis pengajuan izin kerja Tenaga Teknis Kefarmasian *to-be*

4.1.3.6 Proses bisnis penanganan permohonan izin praktik Dokter *to-be*

Pada usulan proses bisnis pengajuan izin praktik untuk Dokter, masih melibatkan 4 pegawai dari sisi Dinas Kesehatan yaitu Petugas Proses; Kepala Seksi SDM; Kepala Bidang SDM, Kefarmasian & Alat Kesehatan; serta Kepala Dinas. Proses bisnis usulan untuk 16 Tenaga Kesehatan yang memiliki kesamaan proses bisnis ini juga dimodelkan menjadi 2 gambar terpisah yaitu proses pengajuan dan penanganan. Proses bisnis penanganan ditunjukkan pada Gambar 4.12 dan penjelasannya sebagai berikut.

1. Setelah Petugas Proses menyetujui berkas permohonan yang diajukan, maka sistem akan mengirimkan pemberitahuan dan data terkait kepada Kepala Seksi SDM
2. Kepala Seksi SDM melaporkan kepada Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan terkait permohonan yang masuk.
3. Setelah melihat berkas yang telah memenuhi persyaratan, Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan menugaskan Petugas Proses untuk membuat Surat Izin Kerja/Praktik tenaga kesehatan.
4. Petugas Proses mencetak surat izin praktik dan salinannya, kemudian meminta persetujuan kepada Kepala Bidang SDM, Kefarmasian & Alat Kesehatan dan Kepala Dinas.
5. Setelah berkas surat disetujui, salinan surat izin akan diarsip kemudian Petugas Proses merubah status penanganan untuk memberitahukan bahwa surat izin telah siap terbit ke Pemohon.
6. Pemohon datang ke Kantor Dinkes membawa hasil cetak tanda bukti penerimaan berkas dan berkas persyaratan untuk mendapatkan surat izin praktik yang telah siap terbit.
7. Petugas proses menambahkan tanda bukti terima berkas dari pemohon ke arsip.



Gambar 4.12 Proses bisnis pengajuan izin praktik Dokter *to-be*

Pada Tabel 4.4 dijelaskan perubahan aktivitas penanganan permohonan izin kerja/praktik untuk seluruh Tenaga Kesehatan. Tenaga Kesehatan berjenis Apoteker, Bidan dan Perawat, memiliki alur proses yang sama, namun terdapat perbedaan pada Kepala Seksi yang bertanggungjawab dan petugas yang terlibat saat pelaksanaan kunjungan survei. Untuk Tenaga Kesehatan berjenis Dokter dan Tenaga Teknis Kefarmasian, juga memiliki proses bisnis yang sama, keduanya tidak memerlukan kunjungan survei dan hanya dibedakan oleh Kepala Seksi yang bertanggungjawab.

Tabel 4.4 Perubahan aktivitas proses bisnis penanganan permohonan izin praktik/kerja Tenaga Kesehatan

Unit	Proses Bisnis As-Is	Proses Bisnis To-Be	Keterangan
Petugas Proses	Melaporkan permohonan masuk langsung ke ruangan Kepala Seksi.	Sistem secara otomatis mengirimkan permohonan yang telah disetujui oleh Petugas Proses.	Diubah.
	Melaporkan permohonan yang telah disetujui Kepala Seksi ke Kepala Bidang.	Sistem secara otomatis mengirimkan permohonan yang telah disetujui oleh Kepala Seksi.	Diubah.
Kepala Bidang	Membuat jadwal pada buku catatan jadwal kunjungan. (berlaku untuk permohonan dari Apoteker, Bidan & Perawat)	Sistem menyediakan formulir untuk membuat jadwal kunjungan. (berlaku untuk permohonan dari Apoteker, Bidan & Perawat)	Diubah.
Kepala Seksi (Kepala Seksi Kefarmasian untuk Apoteker dan Tenaga Teknis Kefarmasian (Kepala Seksi SDM untuk Bidan, Dokter & Perawat)	Membuat berita acara kunjungan pada formulir berita acara. (berlaku untuk permohonan dari Apoteker, Bidan & Perawat)	Sistem menyediakan formulir berita acara. (berlaku untuk permohonan dari Apoteker, Bidan & Perawat)	Diubah.

4.1.4 Analisis perbaikan proses bisnis

Pada Tabel 4.5 dijelaskan mengenai proses bisnis *as-is*, proses bisnis *to-be* jika terdapat perubahan, serta waktu yang dibutuhkan untuk menjalankan setiap *task*. Kolom “Proses Bisnis *As-Is*” berisikan proses bisnis yang sedang berjalan pada saat ini. Sedangkan kolom “Proses Bisnis *To-Be*” berisikan proses bisnis usulan yang diberikan guna menyelesaikan permasalahan yang muncul pada proses bisnis *as-is*. Kolom “Waktu Baru” merupakan waktu yang dibutuhkan untuk mengerjakan *task* pada proses bisnis *as-is*, sedangkan “Perkiraan Waktu Baru” merupakan perkiraan waktu untuk mengerjakan setiap *task* pada proses bisnis usulan (*to-be*). Terdapat beberapa proses bisnis *as-is* yang tidak memiliki proses bisnis usulan (*to-be*) dikarenakan regulasi dari organisasi yang tidak dapat dirubah atau digantikan.

Tabel 4.5 Analisis perbedaan proses bisnis *as-is* dan *to-be* serta waktu yang dibutuhkan pada pengerjaan *task*

No.	Proses Bisnis <i>As-Is</i>	Proses Bisnis <i>To-Be</i>	Waktu Lama	Perkiraan Waktu Baru
1.	Penyerahan berkas permohonan ke Petugas Proses di Kantor Dinkes.	Mengajukan permohonan secara mandiri melalui formulir yang disediakan sistem.	10 menit	15 menit
2.	Memeriksa kelengkapan berkas.	Memeriksa kelengkapan berkas melalui sistem.	20 menit	5 menit
3.	Pembuatan tanda bukti terima berkas dan penyampaian tahapan penanganan yang akan dilalui.	Sistem secara otomatis mengirimkan tanda bukti terima berkas ketika Petugas Proses menyetujui permohonan.	10 menit	5 detik
4.	Persetujuan tahap berikutnya oleh Kepala Seksi.	Melakukan persetujuan melalui sistem.	15 menit	5 menit
5.	Penjadwalan kunjungan survei/visitasi.	Melakukan penjadwalan melalui sistem yang dapat memberikan pemberitahuan batas kunjungan	30 menit	10 menit

Tabel 4.5 Analisis perbedaan proses bisnis *as-is* dan *to-be* serta waktu yang dibutuhkan pada pengerjaan *task*

No.	Proses Bisnis <i>As-Is</i>	Proses Bisnis <i>To-Be</i>	Waktu Lama	Perkiraan Waktu Baru
		pada hari yang dipilih.		
6.	Kunjungan survei/visitasi.	-	1 jam	-
7.	Pembuatan laporan berita acara.	Pembuatan laporan berita acara pada sistem.	1 jam	45 menit
8.	Penentuan lolos atau tidaknya kunjungan yang telah dilakukan.	-	15 menit	-
9.	Penandatanganan Salinan surat izin oleh Kepala Bidang.	-	5 menit	-
10.	Penandatanganan surat izin oleh Kepala Dinas.	-	5 menit	-
11.	Pengarsipan berkas pemohon.	-	10 menit	-

4.1.2 Analisis kebutuhan pemangku kepentingan

Analisis terhadap pemangku kepentingan yang terlibat dalam proses bisnis dilakukan untuk mengidentifikasi tipe-tipe pemangku kepentingan, peran pada penggunaan sistem, serta kebutuhan yang diharapkan pemangku kepentingan dan pengguna untuk ada didalam sistem yang akan dibangun. Berikut adalah jalannya aktivitas-aktivitas tersebut.

1. Tipe pemangku kepentingan

Analisis terhadap tipe pemangku kepentingan digunakan untuk mengetahui karakteristik dan hubungan setiap pemangku kepentingan yang terlibat dalam penggunaan sistem. Tabel 4.6 menunjukkan analisa terhadap tipe pemangku kepentingan dalam SINKES.

Tabel 4.6 Tipe pemangku kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
Pengguna (<i>End-user</i>)		Tenaga kesehatan :

Tabel 4.6 Tipe pemangku kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
	Merupakan pemangku kepentingan yang akan menggunakan sistem secara langsung. Pemangku kepentingan yang masuk kedalam tipe ini akan digambarkan sebagai aktor dalam pemodelan <i>use-case</i> .	Apoteker, Bidan, Perawat, Tenaga Teknis Kefarmasian, Dokter Dinas Kesehatan Kota XYZ: Petugas Proses, Kepala Seksi SDM, Kepala Seksi Kefarmasian, Seksi Kesehatan Keluarga dan Gizi, UPT Puskesmas Wilayah, Kepala Bidang SDM, Kefarmasian & Alat Kesehatan, Kepala Dinas.
Pengembang (<i>Developer</i>)	Merupakan organisasi atau individu yang pada proses produksi (pengembangan) SINKES.	Peneliti.
Pihak yang berwenang	Merupakan organisasi yang memiliki wewenang untuk memberikan informasi yang dibutuhkan untuk pengembangan SINKES.	CV. Sarana Utama Solusindo.

1. Kebutuhan pemangku kepentingan dan pengguna

Analisa kebutuhan pemangku kepentingan dan pengguna dilakukan untuk mengetahui hal-hal yang diharapkan pemangku kepentingan dan pengguna ada dalam sistem yang akan dibangun. Selain itu, dijelaskan pula situasi yang saat ini berjalan dan solusi yang ditawarkan untuk dapat memenuhi kebutuhan yang muncul. Tabel 4.7 menunjukkan hasil analisa kebutuhan pemangku kepentingan dan pengguna SINKES.

Tabel 4.7 Kebutuhan pemangku kepentingan dan pengguna

Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
Memberikan kemudahan Tenaga Kesehatan untuk melakukan pengajuan permohonan surat izin kerja/praktik	Tenaga Kesehatan	Tenaga Kesehatan datang langsung ke Kantor Dinas Kesehatan Kota XYZ dengan membawa berkas persyaratan.	Terdapat fitur pengajuan untuk memudahkan Tenaga Kesehatan dalam mengajukan permohonan dan memberikan berkas persyaratan.

Tabel 4.7 Kebutuhan pemangku kepentingan dan pengguna

Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
Memberikan kemudahan persetujuan pada permohonan yang masuk.	Petugas Proses, Kepala Seksi SDM, Kepala Seksi Kefarmasian	Setelah petugas proses menerima dan memeriksa kelengkapan berkas, berkas yang dinyatakan memenuhi persyaratan dibawa ke ruangan masing-masing Kepala Seksi.	Terdapat fitur persetujuan berkas pemohon yang memungkinkan informasi terkait berkas persyaratan pemohon diteruskan melalui sistem setelah Petugas Proses memberikan persetujuan awal.
Kemudahan dalam memberikan informasi	Kepala Bidang SDM, Kefarmasian & Alat; Petugas Proses; Pemohon	Petugas Proses memberitahukan waktu perkiraan survei dilakukan saat pemohon datang mengajukan permohonan. Petugas proses menelpon setelah penjadwalan ulang survei selesai dilakukan.	Pemohon mendapatkan segala pemberitahuan terkait penanganan permohonan melalui sistem.
Kemudahan dalam memberikan penugasan kunjungan <i>survei</i> .	Kepala Bidang SDM, Kefarmasian & Alat Kesehatan; Petugas Survei	Kepala Bidang SDM, Kefarmasian & Alat Kesehatan memberikan surat tugas kepada Petugas Survei secara langsung.	Surat izin dapat diberikan melalui sistem.
Kemudahan dalam menulis berita acara	Kepala Seksi	Petugas survey membawa kertas berita acara saat melakukan <i>survei</i> .	Petugas Survei dapat menuliskan berita acara di sistem.
Kemudahan dalam melakukan pengarsipan berkas pemohon	Petugas Proses	Petugas Proses mengarsip berkas fisik pemohon tanpa memiliki cadangan data terkait.	Data terkait berkas pemohon disimpan oleh sistem ketika melakukan pengajuan.
Kebutuhan monitoring kinerja	Kepala Dinas	Tahap proses penanganan permohonan surat	Kepala Dinas dapat melihat tahapan proses penanganan

Tabel 4.7 Kebutuhan pemangku kepentingan dan pengguna

Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
		izin tidak tergambar dengan jelas.	permohonan pada sistem.

4.2 User Perspective

Analisa *user perspective* dilakukan untuk mengetahui aktor yang terlibat dalam penggunaan sistem, fitur yang harus ada, pendefinisian kebutuhan fungsional maupun non-fungsional dari SINKES. Kegiatan ini dilakukan untuk mengetahui hal-hal yang digunakan sebagai dasar pembangunan SINKES.

4.2.1 Identifikasi Aktor

Pada tahap ini, dilakukan identifikasi mengenai aktor yang terlibat dalam sistem berdasarkan hasil Analisis Peran dan Pemangku Kepentingan pada Tabel 4.7 bagian Pengguna. Hasil identifikasi aktor pada SINKES ditunjukkan oleh Tabel 4.8.

Tabel 4.8 Tabel identifikasi aktor

Aktor	Deskripsi
User	Setiap pengguna sistem yang belum teridentifikasi otoritasnya.
Tenaga Kesehatan	Setiap orang yang dapat menggunakan sistem untuk mengajukan permohonan, mendapatkan jadwal kunjungan survei dan pemberitahuan terkait penanganan izin yang diajukan.
Petugas Proses	Setiap orang yang dapat menggunakan sistem untuk menerima permohonan, melakukan verifikasi data pemohon dan mendapatkan tugas kunjungan survei.
Kepala Seksi Kefarmasian	Seseorang yang dapat menggunakan sistem untuk melihat data berkas permohonan dari Tenaga Kesehatan dibawah Seksi Kefarmasian yang dikepalai, mendapatkan tugas kunjungan survei dan membuat berita acara survei.
Kepala Seksi SDM	Seseorang yang dapat menggunakan sistem untuk melihat data berkas permohonan dari Tenaga Kesehatan dibawah Seksi SDM yang dikepalai, mendapatkan tugas kunjungan survei dan membuat berita acara survei.
UPT Puskesmas Wilayah	Setiap orang perwakilan dari UPT Puskesmas Wilayah tempat Bidan praktik yang menggunakan sistem untuk mendapatkan tugas kunjungan survei.

Tabel 4.8 Tabel identifikasi aktor

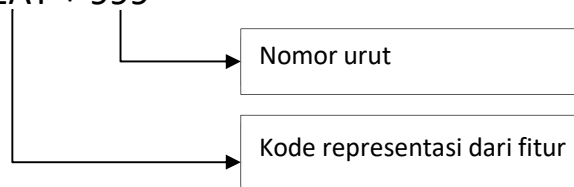
Aktor	Deskripsi
Seksi Kesehatan Keluarga dan Gizi	Setiap orang perwakilan dari Seksi Kesehatan Keluarga yang menggunakan sistem untuk mendapatkan tugas kunjungan survei.
Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan	Seseorang yang dapat menggunakan sistem untuk menjadwalkan dan menugaskan kunjungan <i>survei</i> , menerima berita acara hasil <i>survei</i> .
Kepala Dinas	Seseorang yang dapat menggunakan sistem untuk mengetahui kinerja pegawai Dinkes dalam menangani permohonan surat izin.

Pada Tabel 4.8 ditambahkan identifikasi aktor berjenis *User* yang merupakan pengguna sistem yang belum teridentifikasi otoritasnya dalam sistem. Kemudian aktor Tenaga Kesehatan merupakan bentuk abstraksi pemangku kepentingan Apoteker, Bidan, Dokter, Perawat dan Tenaga Teknis Kefarmasian. Abstraksi tersebut dilakukan karena Apoteker, Bidan, Dokter, Perawat dan Tenaga Teknis Kefarmasian memiliki interaksi yang sama dengan sistem. Sedangkan Petugas Proses; Kepala Seksi Kefarmasian; Kepala Seksi SDM; UPT Puskesmas Wilayah; Seksi Kesehatan Keluarga dan Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; dan Kepala Dinas merupakan aktor yang berasal dari pemangku kepentingan dari pihak Dinas Kesehatan Kota XYZ. Aktor dari pihak Dinkes dijelaskan satu per satu dikarenakan masing-masing memiliki interaksi yang berbeda dengan sistem.

4.2.2 Identifikasi fitur

Berdasarkan kebutuhan pemangku kepentingan dan pengguna yang dijelaskan dalam Tabel 4.7, maka dilakukan identifikasi terhadap kemampuan yang harus dimiliki sistem untuk memenuhi segala kebutuhan yang telah dijabarkan. Tabel 4.9 berikut adalah daftar hasil identifikasi kemampuan yang diharapkan ada dalam sistem.

Kode : FEAT + 999



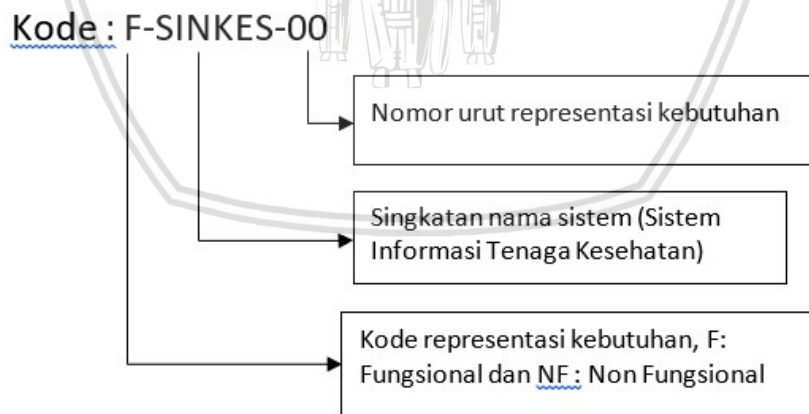
Gambar 4.13 Contoh penomoran fitur dan deskripsinya

Tabel 4.9 Identifikasi fitur

Kode Fitur	Deskripsi	Prioritas
FEAT01	Sistem bisa digunakan untuk mengajukan permohonan surat izin kerja/praktik.	Mo
FEAT02	Sistem bisa digunakan untuk menangani permohonan yang telah diajukan oleh Tenaga Kesehatan.	Mo
FEAT03	Sistem dapat menunjukkan grafik statistik permohonan yang diajukan serta penanganan yang diberikan.	C
FEAT04	Sistem dapat memberikan otoritas pada pengguna sesuai dengan peran dan tanggungjawab yang dimiliki.	Mo
FEAT05	Sistem dapat ditampilkan dengan baik di berbagai ukuran layar perangkat.	S
FEAT06	Sistem dapat berjalan dengan baik pada berbagai jenis <i>browser</i> .	S

4.3 Spesifikasi kebutuhan pengembangan

Berdasarkan hasil identifikasi kebutuhan pemangku kepentingan dan pengguna yang berada pada Tabel 4.7 dan identifikasi fitur yang berada pada Tabel 4.9, maka di tahap berikut ini dilakukan identifikasi kebutuhan sistem berupa kebutuhan fungsional dan non fungsional untuk menunjukkan secara jelas hal-hal yang harus dilakukan oleh sistem.



Gambar 4.14 Contoh penomoran kebutuhan fungsional maupun non fungsional dan deskripsi tiap penomoran

4.3.1 Kebutuhan fungsional

Seperti yang telah dijelaskan pada Bab 2 bagian 2.5 yang membahas tentang *user perspective*, pada tahap ini akan dilakukan identifikasi kebutuhan yang

mengacu pada Tabel 4.7 dan Tabel 4.9. Pada Tabel 4.10 dijelaskan identifikasi kebutuhan fungsional yang dilakukan.



Tabel 4.10 Identifikasi kebutuhan fungsional

Kode Fitur	Kode Kebutuhan Fungsional	Deskripsi
FEAT01	F-SINKES-01	Sistem menyediakan fungsi untuk merubah atau melengkapi data profil diri.
	F-SINKES-02	Sistem menyediakan fungsi untuk merubah atau melengkapi foto diri sebagai salah satu persyaratan yang dibutuhkan untuk mengajukan permohonan surat izin.
	F-SINKES-03	Sistem menyediakan fungsi untuk membuat berkas pendukung yang <i>template</i> -nya telah disediakan oleh pihak Dinkes untuk memudahkan Tenaga Kesehatan.
	F-SINKES-04	Sistem harus menyediakan fungsi mengajukan permohonan surat izin kerja/praktik untuk memfasilitasi Tenaga Kesehatan mengirimkan permohonan serta berkas persyaratan yang diperlukan.
	F-SINKES-05	Sistem menyediakan fungsi untuk memfasilitasi tenaga kesehatan melakukan pengecekan terhadap alur penganangan permohonan yang telah diajukan sebelumnya.
	F-SINKES-06	Sistem menyediakan fungsi untuk melakukan cetak tanda bukti terima berkas jika permohonan yang diajukan telah disetujui.
FEAT02	F-SINKES-07	Sistem menyediakan fungsi untuk melihat daftar permohonan surat izin sesuai dengan otoritas yang dimiliki oleh pihak Dinkes yang mengakses.
	F-SINKES-08	Sistem menyediakan fungsi untuk melihat detil dari setiap permohonan.
	F-SINKES-09	Sistem menyediakan fungsi untuk melakukan persetujuan terhadap permohonan yang telah diajukan Tenaga Kesehatan.
	F-SINKES-10	Sistem menyediakan fungsi untuk memberikan persetujuan penanganan lebih lanjut.

Tabel 4.10 Identifikasi kebutuhan fungsional

Kode Fitur	Kode Kebutuhan Fungsional	Deskripsi
	F-SINKES-11	Sistem menyediakan fungsi untuk menambahkan jadwal kunjungan survei/visitasi.
	F-SINKES-12	Sistem menyediakan fungsi untuk melihat jadwal kunjungan survei.
	F-SINKES-13	Sistem menyediakan fungsi untuk membuat berita acara hasil kunjungan.
	F-SINKES-14	Sistem menyediakan fungsi untuk memberi instruksi pencetakan surat izin dari Kepala Bidang ke Petugas Proses
	F-SINKES-15	Sistem harus menyediakan fungsi untuk mencetak surat izin kerja/praktik yang siap terbit.
FEAT04	F-SINKES-16	Sistem harus menyediakan fungsi membuat akun untuk mendapatkan hak akses ke dalam sistem.
	F-SINKES-17	Sistem harus mengenali dan memberikan hak akses terhadap pengguna sesuai otorisasinya.
	F-SINKES-18	Sistem harus menghapus otorisasi pengguna saat meninggalkan sistem.

4.3.2 Kebutuhan non-fungsional

Pada Tabel 4.11 berikut merupakan hasil identifikasi kebutuhan non fungsional yang harus ada pada sistem untuk mendukung penggunaan. Seperti yang tertera pada tabel, terdapat dua kebutuhan non fungsional yaitu terkait *responsiveness* dan *compatibility*. SINKES yang diharapkan mampu membantu mengoptimalkan pelayanan Dinkes kepada Tenaga Kesehatan, maka perlu menyediakan sebuah sistem yang tampilannya dapat beradaptasi dengan berbagai ukuran resolusi layar perangkat yang digunakan oleh Tenaga Kesehatan yang ingin mengajukan permohonan surat izin kerja/praktik. Tidak hanya itu, sistem juga diharapkan mampu menjalankan fungsi-fungsi dengan baik diberbagai macam *browser*.

Tabel 4.11 Identifikasi kebutuhan non fungsional

Kode Fitur	Kode Fungsi	Nama Fungsi	Deskripsi
FEAT05	NF-SINKES-01	<i>Responsiveness</i>	Sistem harus dapat menampilkan halaman dengan baik diberbagai ukuran resolusi layar.
FEAT06	NF-SINKES-02	<i>Compatibility</i>	Sistem harus dapat menjalankan fungsinya dengan baik dan benar melalui berbagai <i>browser</i> .

4.3.3 Spesifikasi kebutuhan perangkat instalasi

Kebutuhan perangkat instalasi digunakan untuk mendeskripsikan kebutuhan minimal yang harus dimiliki perangkat untuk menjalankan sistem dengan baik. Kebutuhan perangkat instalasi dalam pengembangan SINKES ditunjukkan pada tabel 4.12.

Tabel 4.12 Identifikasi kebutuhan instalasi

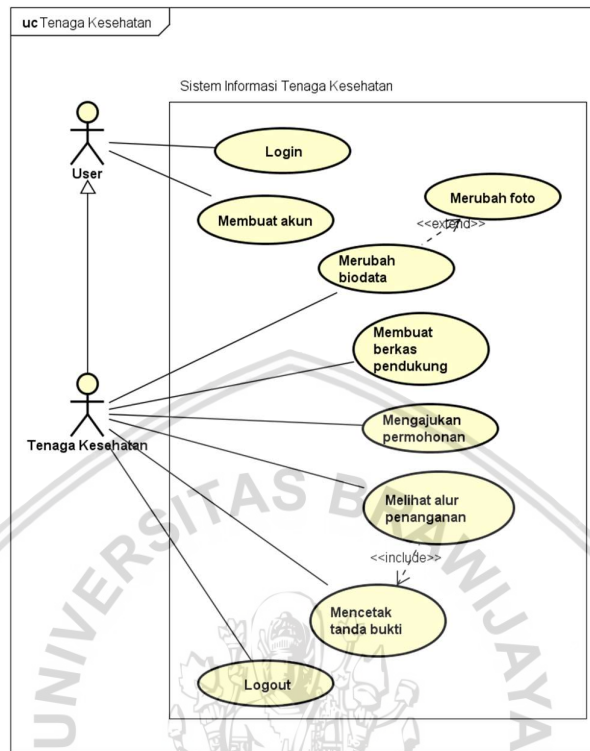
No	Nama Kebutuhan	Versi
1.	<i>Processor</i>	<i>Processor</i> 32-bit (x86) atau 64-bit (x64)
2.	Kapasitas RAM	2048 MB
3.	Sistem Operasi	<i>Windows 7</i>
4.	PHP	7.1.1
5.	<i>MySQL</i>	5.0.12

4.3.4 Use case diagram

Mengacu pada bagian 2.7.1 tentang *use case diagram*, *use case diagram* dibuat dengan menggunakan sudut pandang pengguna ketika berinteraksi dengan sistem. Pemodelan *use case diagram* dilakukan berdasarkan rekomendasi perbaikan proses bisnis (*to-be*) pada bagian 9.1.9 sampai 9.1.14 dan kebutuhan fungsional yang telah didefinisikan sebelumnya pada Tabel 4.10. *Use case diagram* dari sistem yang akan dibangun akan digambarkan menjadi dua buah *use case diagram*, yaitu *use case diagram* untuk kebutuhan sistem dengan aktor Tenaga Kesehatan dan *use case diagram* untuk kebutuhan sistem dengan aktor petugas Dinkes yang terdiri dari Petugas Proses, Kepala Seksi Kefarmasian, Kepala Seksi SDM, Kepala Bidang, Kepala Dinas, perwakilan UPT Puskesmas Wilayah dan Seksi Kesehatan Keluarga & Gizi.

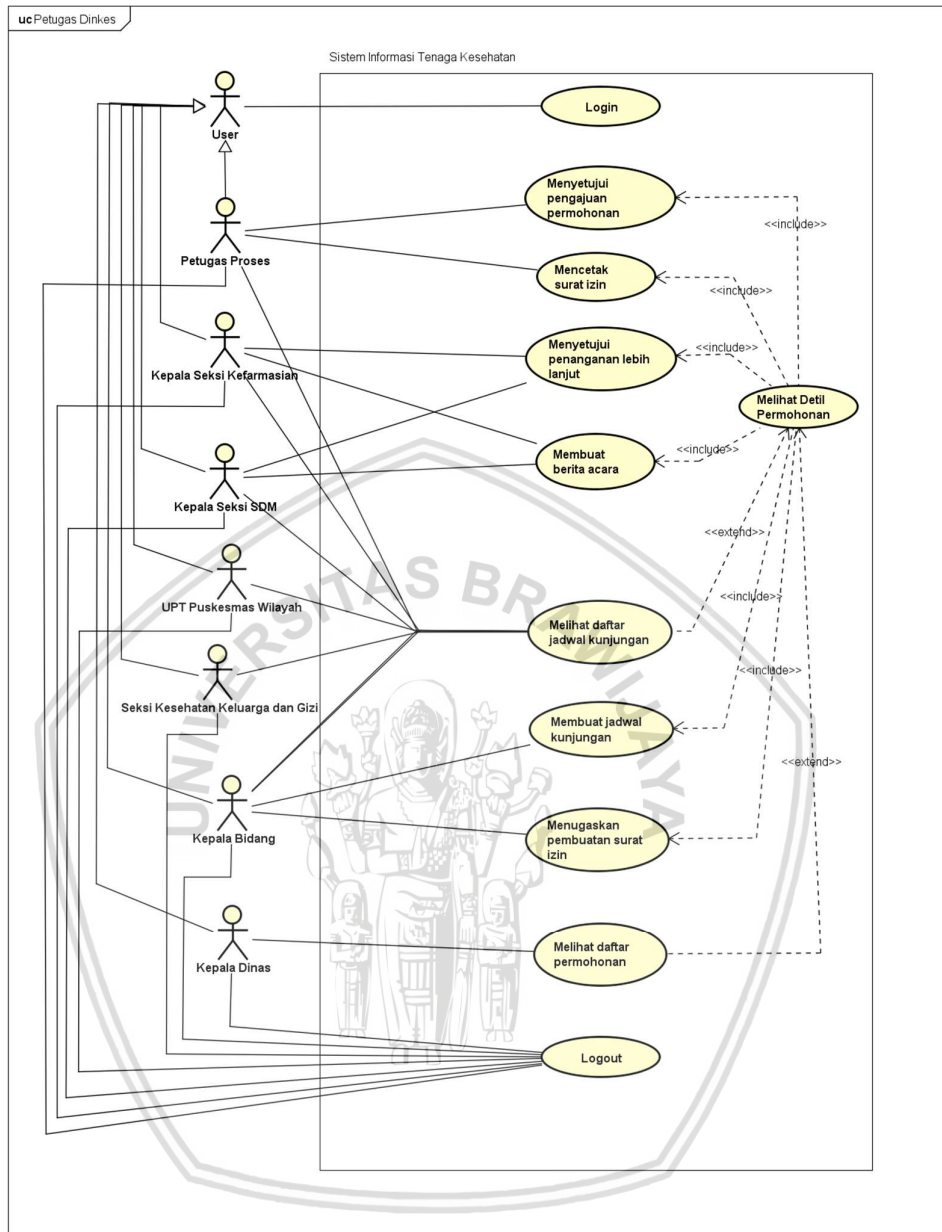
Use case diagram untuk kebutuhan sistem dengan aktor Tenaga Kesehatan akan ditunjukkan dalam Gambar 4.15 berikut, kemudian masing-masing *use case*

yang telah digambarkan pada Gambar 4.15 akan dijelaskan lebih spesifik kedalam tabel *use case scenario* pelapor pada bagian 4.3.6.



Gambar 4.15 Use case diagram Tenaga Kesehatan

Sedangkan *use case diagram* untuk kebutuhan sistem dengan aktor dari petugas Dinkes akan ditunjukkan dalam Gambar 4.16 berikut, dan kemudian masing-masing *use case* yang telah digambarkan dalam Gambar 4.16 akan dijelaskan lebih spesifik kedalam tabel *use case scenario* petugas Dinkes pada bagian 4.3.7.



Gambar 4.16 Use case diagram petugas Dinkes

4.3.5 Use case scenario Tenaga Kesehatan

Use case yang telah digambarkan pada Gambar 4.15 dijelaskan lebih rinci untuk menunjukkan interaksi yang terjadi antara Tenaga Kesehatan dan sistem kedalam tabel *use case scenario*. Tabel tersebut terdiri dari kode *use case* (UCT), kode fungsional yang direferensi (F-SINKES-00), deskripsi, aktor yang terlibat, *scenario* utama dan alternatif, serta kondisi yang harus dipenuhi sebelum *use case* dilakukan dan setelah dilakukan.

4.3.5.1 Use case scenario membuat akun

Pada Tabel 4.13 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* membuat akun.

Tabel 4.13 Use case scenario membuat akun

Kode Use Case	UCT-01
Kode Fungsional	F-SINKES-16
Nama	Membuat akun
Aktor	User
Deskripsi	Use case ini menggambarkan upaya User untuk mendapatkan hak akses masuk kedalam sistem agar dapat mengajukan permohonan surat izin.
Tujuan	Tenaga Kesehatan berhasil mendaftarkan diri.
Kondisi Awal	1. Aktor dalam keadaan tidak <i>login</i> . 2. Halaman <i>login</i> Tenaga Kesehatan ditampilkan pada aktor.
Skenario Utama	
1) User memilih fungsi tampilkan halaman pembuatan akun. 2) Sistem menampilkan halaman berisi formulir pembuatan akun. 3) User memasukkan data yang dibutuhkan berupa nama lengkap, Nomor Induk Kependudukan (NIK), <i>email</i> , nomor <i>handphone</i> , kata sandi, dan jenis kelompok tenaga kesehatan. 4) User memilih fungsi daftar. 5) Sistem menampilkan pesan berhasil mendaftar dan halaman beranda Tenaga Kesehatan.	
Skenario Alternatif	
Jika User mengosongi kolom di formulir : 1) User tidak mengisi formulir secara keseluruhan. 2) Sistem menampilkan pesan peringatan bahwa semua kolom tidak boleh kosong. Jika User mendaftarkan NIK yang pernah didaftarkan sebelumnya : 1) User mendaftarkan diri dengan menggunakan NIK yang telah didaftarkan sebelumnya. 2) Sistem menampilkan pesan peringatan bahwa NIK yang sudah pernah didaftarkan tidak bisa digunakan untuk mendapatkan akun lagi. Jika User mendaftarkan email yang pernah didaftarkan sebelumnya.	

1) <i>User</i> mendaftarkan diri dengan menggunakan <i>email</i> yang telah didaftarkan sebelumnya. 2) Sistem menampilkan pesan peringatan bahwa <i>email</i> yang sudah pernah didaftarkan tidak bisa digunakan untuk mendapatkan akun lagi.	
Kondisi Akhir	<i>User</i> berhasil membuat akun dan berada pada halaman beranda Tenaga Kesehatan.
Ekstensi	-
Sub Aliran	-

4.3.5.2 Use case scenario login

Pada Tabel 4.14 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case Login*.

Tabel 4.14 Use case scenario login

Kode Use Case	UCT-02
Kode Fungsional	F-SINKES-17
Nama	<i>Login</i>
Aktor	<i>User</i>
Deskripsi	<i>Use case</i> ini menggambarkan alur yang dilalui oleh <i>User</i> untuk memasuki sistem sesuai hak akses.
Tujuan	<i>User</i> berhasil masuk ke dalam sistem dan mendapatkan akses sesuai dengan perannya.
Kondisi Awal	1. <i>User</i> mengakses halaman <i>login</i> . 2. <i>User</i> telah memiliki akun. 3. <i>User</i> belum teridentifikasi oleh sistem.
Skenario Utama	
1) <i>User</i> memasukkan NIK dan <i>password</i> . 2) <i>User</i> memilih fungsi <i>login</i> . 3) Sistem melakukan autentifikasi dan menampilkan halaman beranda sesuai peran <i>User</i> .	
Skenario Alternatif	
Jika Tenaga Kesehatan mengisikan NIK dan <i>password</i> yang tidak sesuai : 1) <i>User</i> memasukkan NIK dan <i>password</i> yang tidak sesuai.	

2) Sistem menampilkan pesan peringatan bahwa NIK dan <i>password</i> tidak sesuai.	
Kondisi Akhir	Tenaga kesehatan berhasil membuat akun dan berada pada halaman beranda.
Ekstensi	-
Sub Aliran	-

4.3.5.3 Use case scenario merubah biodata

Pada Tabel 4.15 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* merubah biodata.

Tabel 4.15 Use case scenario merubah biodata

Kode Use Case	UCT-03
Kode Fungsional	F-SINKES-02
Nama	Merubah biodata
Aktor	Tenaga Kesehatan
Deskripsi	<i>Use case</i> ini menggambarkan alur yang dilalui Tenaga Kesehatan untuk merubah/melengkapi data diri sebelum mengajukan permohonan.
Tujuan	Tenaga Kesehatan berhasil merubah data diri.
Kondisi Awal	1. Tenaga Kesehatan dalam status <i>login</i> . 2. Halaman mengubah biodata diri telah ditampilkan kepada Tenaga Kesehatan.
Skenario Utama	
1) Tenaga Kesehatan mengisi/mengubah data yang telah ditampilkan. 2) Tenaga Kesehatan memilih fungsi simpan. 3) Sistem menampilkan pesan bahwa data diri berhasil diubah.	
Skenario Alternatif	
Jika ubah data gagal disimpan sistem : 1) Sistem menampilkan pesan bahwa perubahan gagal disimpan.	
Kondisi Akhir	Sistem menampilkan halaman biodata diri.
Ekstensi	<u>Merubah Foto [UCT-04] :</u>

	<ol style="list-style-type: none"> 1) Tenaga Kesehatan memilih fungsi ganti foto 2) Sistem menampilkan halaman ganti foto 3) Tenaga Kesehatan memilih foto yang diinginkan 4) Tenaga Kesehatan memilih fungsi simpan 5) Sistem menampilkan halaman profil dan pesan berhasil
Sub Aliran	-

4.3.5.4 Use case scenario membuat berkas pendukung

Pada Tabel 4.16 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* membuat berkas pendukung.

Tabel 4.16 Use case scenario membuat berkas pendukung

Kode Use Case	UCT-05
Kode Fungsional	F-SINKES-03
Nama	Membuat berkas pendukung
Aktor	Tenaga Kesehatan
Deskripsi	<i>Use case</i> ini menggambarkan alur yang dilalui Tenaga Kesehatan untuk membuat berkas pendukung yang <i>template</i> -nya telah disediakan oleh Dinkes. Namun Tenaga Kesehatan jenis Perawat tidak bisa melakukan <i>use case</i> ini dikarenakan Dinkes tidak menyediakan <i>template</i> apapun untuk Tenaga Kesehatan tersebut.
Tujuan	Tenaga Kesehatan berhasil mendapatkan berkas yang telah dibuat melalui sistem.
Kondisi Awal	<ol style="list-style-type: none"> 1. Tenaga Kesehatan dalam kondisi <i>login</i>. 2. Halaman berisi formulir pembuatan berkas telah ditampilkan.
Skenario Utama	
<ol style="list-style-type: none"> 1) Tenaga Kesehatan memasukkan data kedalam formulir. 2) Tenaga Kesehatan memilih fungsi unduh berkas. 	

3) Sistem mengunduh berkas yang telah dibuat.	
Skenario Alternatif	
-	
Kondisi Akhir	Tenaga Kesehatan mendapatkan berkas yang telah dibuat.
Ekstensi	-
Sub Aliran	-

4.3.5.5 Use case scenario mengajukan permohonan

Pada Tabel 4.17 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* mengajukan permohonan.

Tabel 4.17 Use case scenario mengajukan permohonan

Kode Use Case	UCT-06
Kode Fungsional	F-SINKES-04
Nama	Mengajukan permohonan
Aktor	Tenaga Kesehatan
Deskripsi	Use case ini menggambarkan Tenaga Kesehatan ketika melakukan pengajuan permohonan surat izin sesuai dengan kelompok profesinya.
Tujuan	Tenaga Kesehatan berhasil mengajukan permohonan surat izin.
Kondisi Awal	<ol style="list-style-type: none"> 1. Tenaga Kesehatan dalam kondisi <i>login</i>. 2. Tenaga Kesehatan belum melakukan pengajuan sebelumnya, atau sudah mengajukan namun berkas ditolak oleh petugas di Dinkes. 3. Halaman pengajuan telah ditampilkan kepada Tenaga Kesehatan
Skenario Utama	
<ol style="list-style-type: none"> 1) Tenaga Kesehatan mengisi formulir sesuai dengan data yang dibutuhkan untuk setiap jenis Tenaga Kesehatan. 2) Tenaga Kesehatan memilih fungsi ajukan permohonan untuk mengirimkan formulir yang telah diisi. 	

3) Sistem kembali menampilkan halaman pengajuan dan pesan bahwa data berhasil dikirimkan.	
Skenario Alternatif	
1) Jika permohonan gagal diajukan, maka sistem menampilkan pesan error.	
Kondisi Akhir	Sistem menampilkan halaman pengajuan dan pesan bahwa data berhasil dikirimkan.
Ekstensi	-
Sub Aliran	-

4.3.5.6 Use case scenario melihat alur penanganan

Pada Tabel 4.18 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* melihat alur penanganan.

Tabel 4.18 Use case scenario melihat alur penanganan

Kode Use Case	UCT-07
Kode Fungsional	F-SINKES-05
Nama	Melihat alur penanganan
Aktor	Tenaga Kesehatan
Deskripsi	<i>Use case</i> ini menggambarkan Tenaga Kesehatan dalam melihat alur penanganan permohonan yang telah diajukan untuk mengetahui tahap yang telah dilalui.
Tujuan	Tenaga Kesehatan berhasil melihat informasi penanganan permohonan dari Dinkes.
Kondisi Awal	1. Tenaga Kesehatan dalam kondisi <i>login</i> . 2. Tenaga Kesehatan telah mengajukan permohonan sebelumnya.
Skenario Utama	
1) Tenaga Kesehatan memilih fungsi untuk menampilkan halaman beranda. 2) Sistem menampilkan halaman beranda yang terdapat tabel penanganan yang telah dilalui oleh permohonan yang diajukan.	
Skenario Alternatif	
1) Tenaga Kesehatan memilih fungsi untuk menampilkan halaman beranda ketika belum mengajukan permohonan.	

2) Sistem menampilkan halaman beranda dengan tabel yang tidak terdapat informasi mengenai penanganan Dinkes.	
Kondisi Akhir	Sistem menampilkan halaman beranda yang terdapat tabel penanganan yang telah dilalui oleh permohonan yang diajukan.
Ekstensi	-
Sub Aliran	-

4.3.5.7 Use case scenario mencetak tanda bukti

Pada Tabel 4.19 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* mencetak tanda bukti.

Tabel 4.19 Use case scenario mencetak tanda bukti

Kode Use Case	UCT-08
Kode Fungsional	F-SINKES-06
Nama	Mencetak tanda bukti
Aktor	Tenaga Kesehatan
Deskripsi	<i>Use case</i> ini menggambarkan Tenaga Kesehatan dalam mencetak tanda bukti terima berkas setelah permohonan yang diajukan diterima oleh pihak Dinkes.
Tujuan	Tenaga Kesehatan mencetak tanda bukti.
Kondisi Awal	<ol style="list-style-type: none"> 1. Tenaga Kesehatan dalam kondisi <i>login</i>. 2. Permohonan yang diajukan disetujui untuk mendapatkan penanganan lebih lanjut. 3. Halaman beranda telah ditampilkan kepada Tenaga Kesehatan.
Skenario Utama	
<ol style="list-style-type: none"> 1) Tenaga Kesehatan memilih fungsi untuk mengunduh tanda bukti terima berkas. 2) Sistem mengunduh tanda bukti terima berkas. 	
Skenario Alternatif	
-	
Kondisi Akhir	Tenaga Kesehatan mendapatkan tanda bukti terima berkas.

Ekstensi	-
Sub Aliran	-

4.3.5.8 Use case scenario logout

Pada Tabel 4.20 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case Logout*.

Tabel 4.20 Use case scenario logout

Kode Use Case	UCT-09
Kode Fungsional	F-SINKES-18
Nama	<i>Logout</i>
Aktor	Tenaga Kesehatan
Deskripsi	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor keluar dari sistem dan menghapus otorisasinya
Tujuan	User keluar dari sistem.
Kondisi Awal	User dalam status <i>login</i> .
Skenario Utama	
1) Tenaga Kesehatan memilih fungsi <i>logout</i> .	
2) Sistem mengarahkan user ke halaman <i>login</i> dan menghapus otorisasinya.	
Skenario Alternatif	
-	
Kondisi Akhir	Tenaga Kesehatan kembali ke halaman <i>login</i> .
Ekstensi	-
Sub Aliran	-

4.3.6 Use case scenario Petugas Dinkes

Use case yang telah digambarkan pada Gambar 4.16 dijelaskan lebih rinci untuk menunjukkan interaksi yang terjadi antara petugas Dinkes yang terdiri dari aktor Petugas Proses, Kepala Seksi Kefarmasian, Kepala Seksi SDM, Kepala Bidang, Kepala Dinas, perwakilan UPT Puskesmas Wilayah serta perwakilan Seksi Kesehatan Keluarga dengan sistem kedalam tabel *use case scenario*. Tabel tersebut terdiri dari kode *use case* (UCP), kode fungsional yang direferensi (F-SINKES-00), deskripsi, aktor yang terlibat, *scenario* utama dan alternatif, serta kondisi yang harus dipenuhi sebelum *use case* dilakukan dan setelah dilakukan.

4.3.6.1 Use case scenario Login

Pada Tabel 4.21 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case Login*.

Tabel 4.21 Use case scenario login

Kode Use Case	UCP-01
Kode Fungsional	F-SINKES-17
Nama	<i>Login</i>
Aktor	<i>User</i>
Deskripsi	<i>Use case</i> ini menggambarkan alur yang dilalui oleh <i>User</i> untuk memasuki sistem sesuai hak akses.
Tujuan	<i>User</i> berhasil masuk ke dalam sistem dan mendapatkan akses sesuai dengan perannya.
Kondisi Awal	<ol style="list-style-type: none"> 1. <i>User</i> mengakses halaman <i>login</i>. 2. <i>User</i> telah memiliki akun. 3. <i>User</i> belum teridentifikasi oleh sistem.
Skenario Utama	
<ol style="list-style-type: none"> 1) <i>User</i> memasukkan <i>username</i> dan <i>password</i>. 2) <i>User</i> memilih fungsi <i>login</i>. 3) Sistem melakukan autentifikasi dan menampilkan halaman beranda sesuai peran <i>User</i>. 	
Skenario Alternatif	
<ol style="list-style-type: none"> 1) <i>User</i> memasukkan <i>username</i> dan <i>password</i> yang tidak sesuai. 2) Sistem menampilkan pesan peringatan bahwa <i>username</i> dan <i>password</i> tidak sesuai. 	
Kondisi Akhir	Sistem menampilkan halaman beranda sesuai dengan peran <i>user</i> .
Ekstensi	-
Sub Aliran	-

4.3.6.2 Use case scenario menyetujui pengajuan permohonan

Pada Tabel 4.22 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* menyetujui pengajuan permohonan.

Tabel 4.22 *Use case scenario* menyetujui pengajuan permohonan

Kode Use Case	UCP-02
Kode Fungsional	F-SINKES-09
Nama	Menyetujui pengajuan permohonan
Aktor	Petugas Proses
Deskripsi	<i>Use case</i> ini menggambarkan Petugas Proses dalam menyetujui permohonan yang diajukan oleh Tenaga Kesehatan (pemohon) untuk ditangani lebih lanjut.
Tujuan	Petugas Proses berhasil menyetujui berkas permohonan.
Kondisi Awal	<ol style="list-style-type: none"> 1. Petugas Proses dalam status <i>login</i>. 2. Adanya permohonan baru yang masuk. 3. Halaman berisi daftar permohonan masuk telah ditampilkan kepada Petugas Proses.
Skenario Utama	
<ol style="list-style-type: none"> 1) Petugas Proses memilih fungsi menampilkan detil berkas permohonan. 2) Sistem menampilkan halaman berisi detil berkas permohonan. 3) Petugas Proses memeriksa kelengkapan berkas yang dilampirkan oleh pemohon. 4) Petugas Proses memilih pilihan menerima permohonan dan mengisi formulir tindakan yang dilakukan, kemudian memilih fungsi kirim. 5) Sistem menampilkan pesan bahwa permohonan telah disetujui. 	
Skenario Alternatif	
<ol style="list-style-type: none"> 1) Petugas Proses memilih menolak permohonan dan mengisi formulir tindakan yang dilakukan, kemudian memilih fungsi kirim. 2) Sistem menampilkan pesan peringatan bahwa permohonan telah ditolak. 	
Kondisi Akhir	Sistem menampilkan pesan bahwa permohonan telah disetujui dan data permohonan yang sedang ditangani bertambah.
Ekstensi	-
Sub Aliran	Melihat Detil Permohonan [UCP-03] <ol style="list-style-type: none"> 1) Aktor memilih fungsi lihat detil pada permohonan yang diinginkan.

	2) Sistem menampilkan halaman detail permohonan yang dipilih.
--	---

4.3.6.3 Use case scenario menyetujui penanganan lebih lanjut

Pada Tabel 4.23 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* menyetujui penanganan lebih lanjut.

Tabel 4.23 Use case scenario menyetujui penanganan lebih lanjut

Kode Use Case	UCP-04
Kode Fungsional	F-SINKES-10
Nama	Menyetujui penanganan lebih lanjut
Aktor	Kepala Seksi Kefarmasian dan Kepala Seksi SDM.
Deskripsi	<i>Use case</i> ini menggambarkan Kepala Seksi Kefarmasian dan Kepala Seksi SDM menyetujui permohonan untuk mendapatkan penanganan selanjutnya oleh Kepala Bidang. Berkas permohonan yang disetujui sesuai dengan seksi yang masing-masing ketua.
Tujuan	Permohonan dari Tenaga Kesehatan diteruskan ke Kepala Bidang.
Kondisi Awal	1. Kepala Seksi dalam status <i>login</i> . 2. Kepala Seksi berada pada halaman detail permohonan.
Skenario Utama	
1) Kepala Seksi mengisi formulir tindakan. 2) Kepala Seksi memilih fungsi kirim. 3) Sistem menampilkan pesan berhasil disimpan.	
Skenario Alternatif	
-	
Kondisi Akhir	Sistem menampilkan pesan berhasil disimpan.
Ekstensi	-
Sub Aliran	Melihat Detil Permohonan [UCP-03] 1) Aktor memilih fungsi lihat detil pada permohonan yang diinginkan.

	2) Sistem menampilkan halaman detail permohonan yang dipilih.
--	---

4.3.6.4 Use case scenario membuat jadwal kunjungan

Pada Tabel 4.24 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* membuat jadwal kunjungan.

Tabel 4.24 Use case scenario membuat jadwal kunjungan

Kode Use Case	UCP-05
Kode Fungsional	F-SINKES-11
Nama	Membuat jadwal kunjungan
Aktor	Kepala Bidang
Deskripsi	<i>Use case</i> ini menggambarkan Kepala Bidang melakukan penjadwalan kunjungan sebelum memeriksa kelayakan tempat praktik Tenaga Kesehatan
Tujuan	Kepala Bidang berhasil membuat jadwal kunjungan.
Kondisi Awal	<ol style="list-style-type: none"> 1. Kepala Bidang dalam status <i>login</i>. 2. Berkas dinyatakan sesuai persyaratan oleh Kepala Seksi. 3. Membuka halaman detail permohonan tertentu.
Skenario Utama	
<ol style="list-style-type: none"> 1) Kepala Bidang mengisi data yang dibutuhkan pada formulir pembuatan jadwal. 2) Kepala Bidang memilih fungsi simpan. 3) Sistem menampilkan pesan berhasil disimpan. 	
Skenario Alternatif	
<p>Jika Kepala Bidang mengosongi kolom pada formulir :</p> <ol style="list-style-type: none"> 1) Sistem menampilkan pesan bahwa kolom tidak boleh dikosongi. <p>Jika jadwal yang dibuat telah melebihi kuota pada hari yang dipilih :</p> <ol style="list-style-type: none"> 1) Sistem menampilkan pesan gagal menyimpan dan kembali pada formulir pembuatan jadwal. 	
Kondisi Akhir	Sistem menampilkan pesan berhasil disimpan.
Ekstensi	-

Sub Aliran	Melihat Detil Permohonan [UCP-03] 1) Aktor memilih fungsi lihat detil pada permohonan yang diinginkan. 2) Sistem menampilkan halaman detil permohonan yang dipilih.
-------------------	---

4.3.6.5 Use case scenario melihat jadwal kunjungan

Pada Tabel 4.25 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* melihat jadwal kunjungan.

Tabel 4.25 Use case scenario melihat jadwal kunjungan

Kode Use Case	UCP-06
Kode Fungsional	F-SINKES-12
Nama	Melihat jadwal kunjungan
Aktor	Petugas Proses, Kepala Seksi Kefarmasian, Kepala Seksi SDM, UPT Puskesmas Wilayah, Seksi Kesehatan Keluarga & Gizi, Kepala Dinas
Deskripsi	<i>Use case</i> ini menggambarkan aktor ketika melihat jadwal kunjungan.
Tujuan	Aktor berhasil melihat jadwal kunjungan.
Kondisi Awal	1. Aktor dalam status <i>login</i> . 2. Jadwal kunjungan selesai dibuat.
Skenario Utama	
1) Aktor memilih fungsi menampilkan halaman jadwal kunjungan. 2) Sistem menampilkan halaman berisi jadwal kunjungan. 3) Aktor memilih salah satu jadwal yang ada. 4) Sistem menampilkan detil informasi mengenai jadwal yang dipilih.	
Skenario Alternatif	
-	
Kondisi Akhir	Sistem menampilkan halaman berisi jadwal kunjungan.
Ekstensi	-
Sub Aliran	Melihat Detil Permohonan [UCP-03] 1) Aktor memilih fungsi lihat detil pada permohonan yang diinginkan.

	2) Sistem menampilkan halaman detail permohonan yang dipilih.
--	---

4.3.6.6 Use case scenario membuat berita acara

Pada Tabel 4.26 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* membuat berita acara.

Tabel 4.26 Use case scenario membuat berita acara

Kode Use Case	UCP-07
Kode Fungsional	F-SINKES-12
Nama	Membuat berita acara
Aktor	Kepala Seksi Kefarmasian dan Kepala Seksi SDM
Deskripsi	<i>Use case</i> ini menggambarkan Kepala Seksi membuat berita acara pemeriksaan untuk diberikan Kepada Kepala Bidang. <i>Use case</i> ini hanya berlaku untuk permohonan yang dikirimkan oleh Tenaga Kesehatan jenis Apoteker, Bidan dan Dokter.
Tujuan	Kepala Seksi berhasil membuat berita acara kunjungan.
Kondisi Awal	<ol style="list-style-type: none"> 1. Status <i>login</i> sukses sebagai Kepala Seksi. 2. Jadwal kunjungan telah dibuat. 3. Telah melakukan kunjungan ke tempat praktik. 4. Halaman berisi detail permohonan dengan status telah dibuatkan jadwal oleh Kepala Bidang ditampilkan kepada Kepala Seksi.
Skenario Utama	
<ol style="list-style-type: none"> 1) Kepala Seksi mengisi formulir berita acara. 2) Kepala Seksi memilih fungsi kirim. 3) Sistem menampilkan pesan berhasil membuat berita acara. 	
Skenario Alternatif	
<p>Jika Kepala Seksi mengosongi kolom pada formulir.</p> <ol style="list-style-type: none"> 1) Sistem menampilkan pesan peringatan bahwa kolom tidak boleh kosong. 	

Kondisi Akhir	Sistem menampilkan pesan berhasil disimpan.
Ekstensi	-
Sub Aliran	Melihat Detil Permohonan [UCP-03] 1) Aktor memilih fungsi lihat detil pada permohonan yang diinginkan. 2) Sistem menampilkan halaman detil permohonan yang dipilih.

4.3.6.7 Use case scenario menugaskan pembuatan surat izin

Pada Tabel 4.27 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* menugaskan pembuatan surat izin.

Tabel 4.27 Use case scenario menugaskan pembuatan surat izin

Kode Use Case	UCP-08
Kode Fungsional	F-SINKES-14
Nama	Menyetujui pembuatan surat izin
Aktor	Kepala Bidang
Deskripsi	<i>Use case</i> ini menggambarkan Kepala Bidang memberikan persetujuan untuk pembuatan surat izin setelah Kepala Seksi menyetujui berkas.
Tujuan	Kepala Bidang berhasil memberikan persetujuan pembuatan surat izin agar Petugas Proses dapat melakukan pencetakan.
Kondisi Awal	1. Kepala Bidang dalam status <i>login</i> . 2. Berkas disetujui oleh Kepala Seksi dan atau adanya berita acara hasil kunjungan.
Skenario Utama	
<p><u>Untuk Tenaga Kesehatan yang disurvei :</u></p> <ol style="list-style-type: none"> 1) Kepala Bidang memilih fungsi melihat detil salah satu permohonan. 2) Sistem menampilkan halaman berisi detil permohonan dan formulir untuk memberikan instruksi pembuatan surat izin. 3) Kepala Bidang memeriksa hasil penilaian berita acara. 4) Kepala Bidang mengisi formulir instruksi pencetakan surat izin. 	

5) Kepala Bidang memilih fungsi simpan. 6) Sistem menampilkan pesan berhasil menyimpan instruksi. <u>Untuk Tenaga Kesehatan yang tidak disurvei :</u> 1) Kepala Bidang memilih fungsi melihat detail salah satu permohonan. 2) Sistem menampilkan halaman berisi detail permohonan dan formulir untuk memberikan instruksi pembuatan surat izin. 3) Kepala Bidang mengisi formulir instruksi pencetakan surat izin. 4) Kepala Bidang memilih fungsi simpan. 5) Sistem menampilkan pesan berhasil menyimpan instruksi.	
Skenario Alternatif	
Untuk Tenaga Kesehatan yang disurvei : 1) Jika penilaian berita acara tidak memenuhi ketentuan, maka Kepala Bidang memilih fungsi buat jadwal ulang dan menjalankan [UCP-06]. Untuk Tenaga Kesehatan yang tidak disurvei : 1) Jika instruksi gagal disimpan, maka sistem menampilkan pesan gagal disimpan.	
Kondisi Akhir	Sistem menampilkan pesan berhasil menyimpan instruksi.
Ekstensi	-
Sub Aliran	Melihat Detail Permohonan [UCP-03] 1) Aktor memilih fungsi lihat detail pada permohonan yang diinginkan. 2) Sistem menampilkan halaman detail permohonan yang dipilih.

4.3.6.8 Use case scenario mencetak surat izin

Pada Tabel 4.28 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case* mencetak surat izin.

Tabel 4.28 Use case scenario mencetak surat izin

Kode Use Case	UCP-09
Kode Fungsional	F-SINKES-15
Nama	Mencetak surat izin
Aktor	Petugas Proses
Deskripsi	<i>Use case</i> ini menggambarkan proses setelah Kepala Bidang menyatakan bahwa menyetujui pembuatan surat izin, maka

	Petugas Proses melakukan pencetakan surat izin kerja/praktik yang nantinya akan diserahkan kepada Tenaga Kesehatan.
Tujuan	Petugas Proses berhasil mencetak dokumen yang telah dihasilkan oleh sistem.
Kondisi Awal	<ol style="list-style-type: none"> 1. Petugas Proses dalam status <i>login</i>. 2. Kepala Bidang menyetujui pembuatan surat izin.
Skenario Utama	
	<ol style="list-style-type: none"> 1) Petugas Proses memilih permohonan yang telah disetujui oleh Kepala Bidang untuk dilakukan pencetakan surat izin. 2) Sistem menampilkan detil permohonan. 3) Petugas Proses memilih fungsi mencetak surat izin. 4) Sistem mengunduh surat izin.
Skenario Alternatif	
	-
Kondisi Akhir	Sistem mengunduh surat izin.
Ekstensi	-
Sub Aliran	Melihat Detil Permohonan [UCP-03] <ol style="list-style-type: none"> 1) Aktor memilih fungsi lihat detil pada permohonan yang diinginkan. 2) Sistem menampilkan halaman detil permohonan yang dipilih.

4.3.6.9 Use case scenario logout

Pada Tabel 4.29 berikut ini akan dijelaskan mengenai deskripsi lengkap dari *use case logout*.

Tabel 4.29 Use case scenario logout

Kode Use Case	UCP-10
Kode Fungsional	F-SINKES-18
Nama	<i>Logout</i>
Aktor	Petugas Proses, Kepala Seksi Kefarmasian, Kepala Seksi SDM, UPT Puskesmas Wilayah, Seksi Kesehatan Keluarga & Gizi, Kepala Dinas.

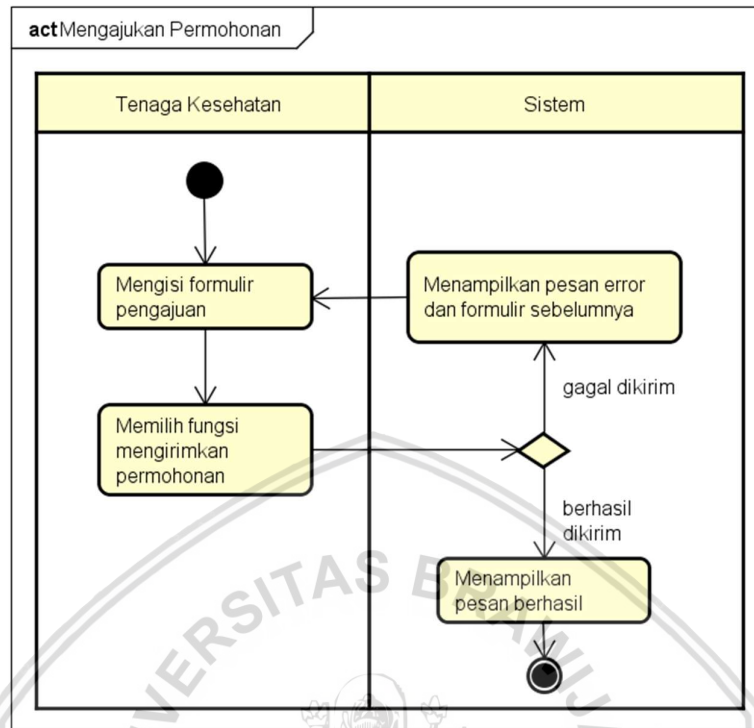
Deskripsi	<i>Use case</i> ini mendeskripsikan tentang bagaimana aktor keluar dari sistem dan menghapus otorisasinya
Tujuan	<i>User</i> keluar dari sistem.
Kondisi Awal	<i>User</i> dalam status <i>login</i> .
Skenario Utama	
1) Aktor memilih fungsi <i>logout</i> . 2) Sistem mengarahkan <i>user</i> ke halaman <i>login</i> dan menghapus otorisasinya.	
Skenario Alternatif	
-	
Kondisi Akhir	Aktor kembali ke halaman <i>login</i> .
Ekstensi	-
Sub Aliran	-

4.4 Activity diagram

Activity diagram pada bagian ini digunakan untuk menggambarkan alur aktifitas aktor dan sistem dari *use case* yang telah dijabarkan pada sub bab 4.3.6 dan 4.3.7. Aktivitas yang dimodelkan berikut merupakan fungsional-fungsional utama yang dapat mewakili gambaran sistem secara garis besar.

4.4.1 Activity diagram mengajukan permohonan

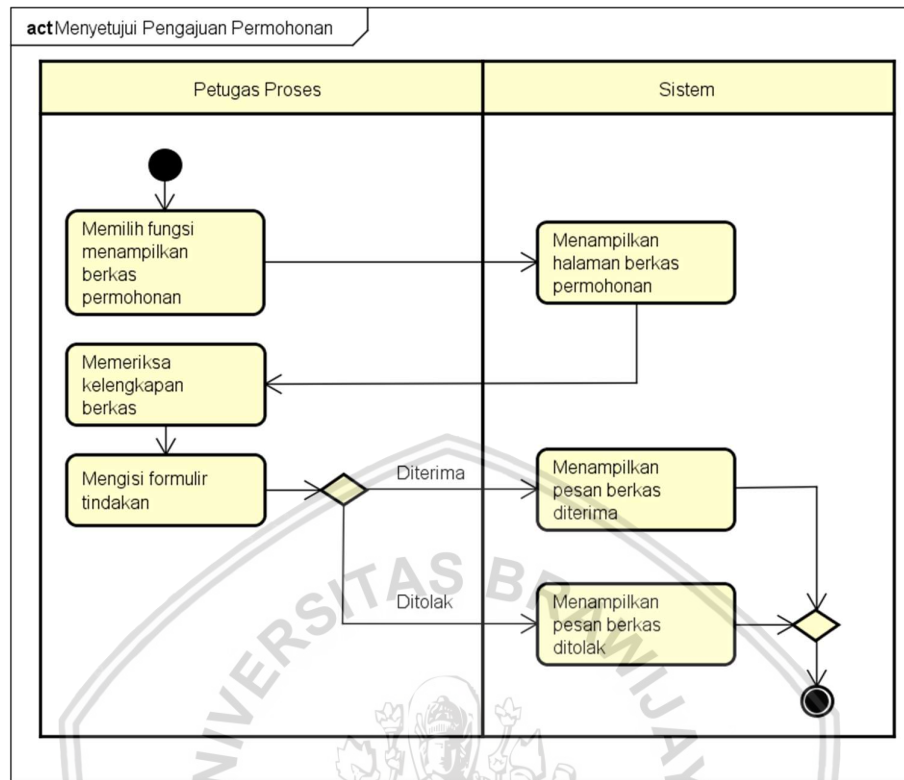
Activity diagram yang ditunjukkan pada Gambar 4.17 merupakan diagram aktivitas untuk *usecase* mengajukan permohonan. *Activity diagram* tersebut menggambarkan alur yang dilakukan oleh Tenaga Kesehatan ketika melakukan pengajuan permohonan surat izin kerja/praktik ke Dinkes Kota XYZ.



Gambar 4.17 Activity Diagram Mengajukan Permohonan

4.4.2 Activity diagram menyetujui pengajuan permohonan

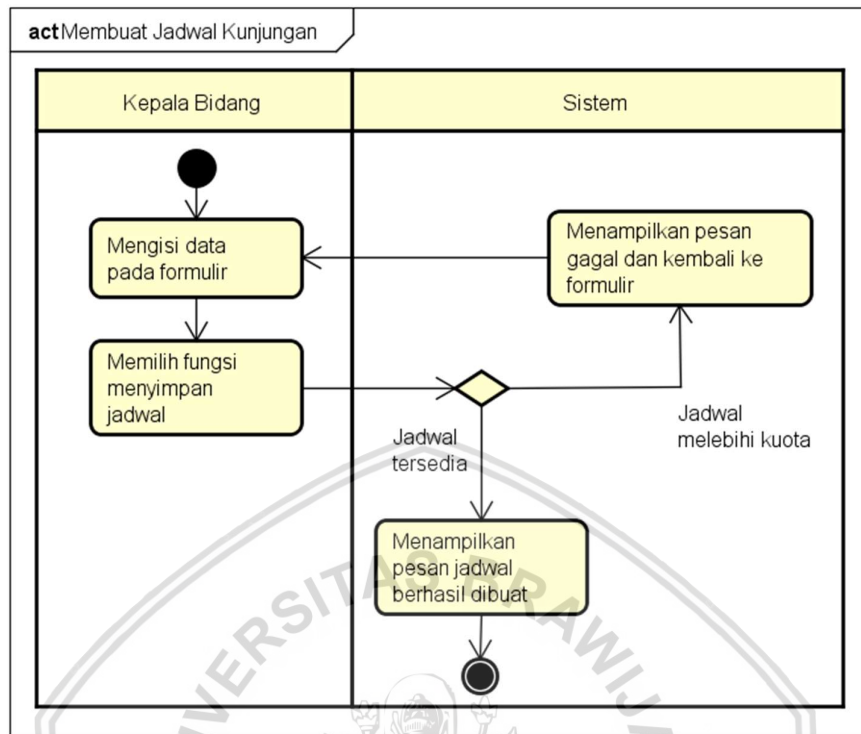
Activity diagram yang ditunjukkan pada Gambar 4.18 merupakan diagram aktivitas untuk *usecase* menyetujui pengajuan permohonan. Activity diagram tersebut menggambarkan alur yang dilakukan oleh Petugas Proses untuk melakukan persetujuan atau penolakan terhadap permohonan yang masuk. Permohonan akan dinyatakan disetujui jika berkas persyaratan yang diajukan telah lengkap dan sesuai.



Gambar 4.18 Activity diagram menyetujui berkas permohonan

4.4.3 Activity Diagram membuat jadwal kunjungan

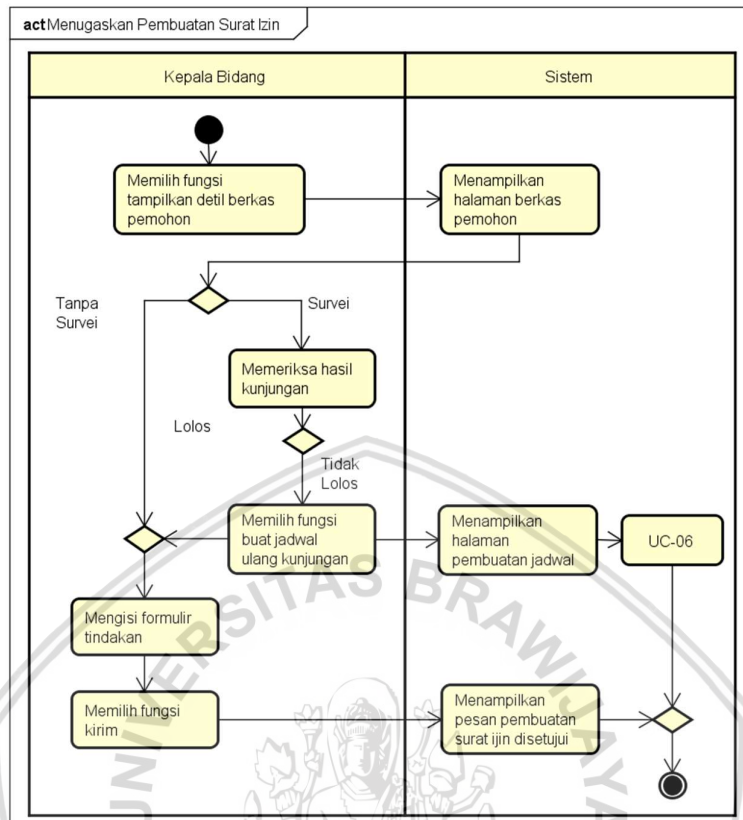
Activity diagram yang ditunjukkan pada Gambar 4.19 merupakan diagram aktivitas untuk *usecase* menambah jadwal kunjungan. Activity diagram tersebut menggambarkan alur yang dilakukan oleh Kepala Bidang untuk membuat jadwal kunjungan survei ke tempat kerja/praktik pemohon untuk menilai kelayakannya. Hasil penilaian yang didapatkan akan berpengaruh terhadap penerbitan surat izin kepada pemohon yang bersangkutan.



Gambar 4.19 Activity diagram menambah jadwal kunjungan

4.4.4 Activity diagram menugaskan pembuatan surat tugas

Activity diagram yang ditunjukkan pada Gambar 4.21 merupakan diagram aktivitas untuk *usecase* menugaskan pembuatan surat izin. Activity diagram tersebut menggambarkan alur yang dilakukan oleh Kepala Bidang untuk memberikan arahan pencetakan surat izin kepada Petugas Proses. Surat tugas akan dicetak jika pemohon telah memenuhi persyaratan yang ada. Untuk pemohon yang disurvei, Kepala Bidang akan mengirimkan arahan pencetakan surat izin kepada Petugas Proses jika penilaian kelayakan tempat kerja/praktik dinyatakan lolos persyaratan.

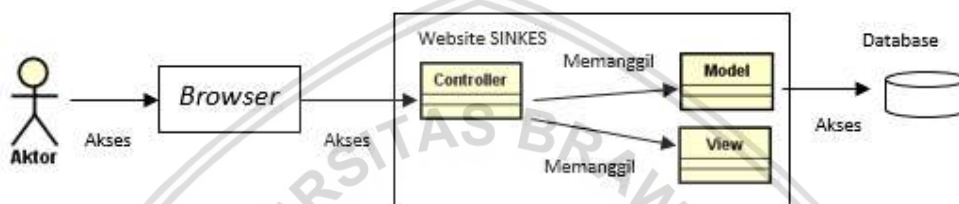


Gambar 4.20 Activity diagram menugaskan pembuatan surat tugas

BAB 5 PERANCANGAN SISTEM

5.1 Perancangan arsitektur sistem

Setelah melakukan analisis dan spesifikasi kebutuhan, maka berdasarkan SDLC *V-Model* aktivitas selanjutnya yang dilakukan adalah mendesain arsitektur sistem untuk menjelaskan komponen struktur utama dari sistem dan hubungan antar komponen-komponen tersebut. Pada bagian ini arsitektur sistem bertujuan untuk menggambarkan bagaimana interaksi antar komponen-komponen SINKES dalam membantu Tenaga Kesehatan dan petugas dari Dinkes Kota XYZ untuk mengajukan dan menangani permohonan surat izin kerja/praktik. Perancangan arsitektur SINKES dapat dilihat pada Gambar 5.1 berikut



Gambar 5.1 Architecture diagram SINKES

Gambar 5.1 menjelaskan bahwa Aktor yang terdiri dari Tenaga Kesehatan dan Petugas Dinkes sama-sama mengakses *website* SINKES untuk melakukan kegiatan pengajuan permohonan bagi Tenaga Kesehatan dan penanganan permohonan bagi Petugas Dinkes. Agar Aktor dapat mengakses *website* SINKES, maka Aktor memerlukan *web browser*. *Website* SINKES dikembangkan dengan menggunakan bahasa pemrograman PHP dan kerangka kerja *Codeigniter* yang menerapkan pola perancangan *Model-View-Controller* (MVC). Akses terhadap *database* dilakukan oleh *Model* yang sebelumnya telah dipanggil oleh *Controller*, kemudian data yang didapatkan akan ditampilkan pada *View* berdasarkan perintah dari *Controller*.

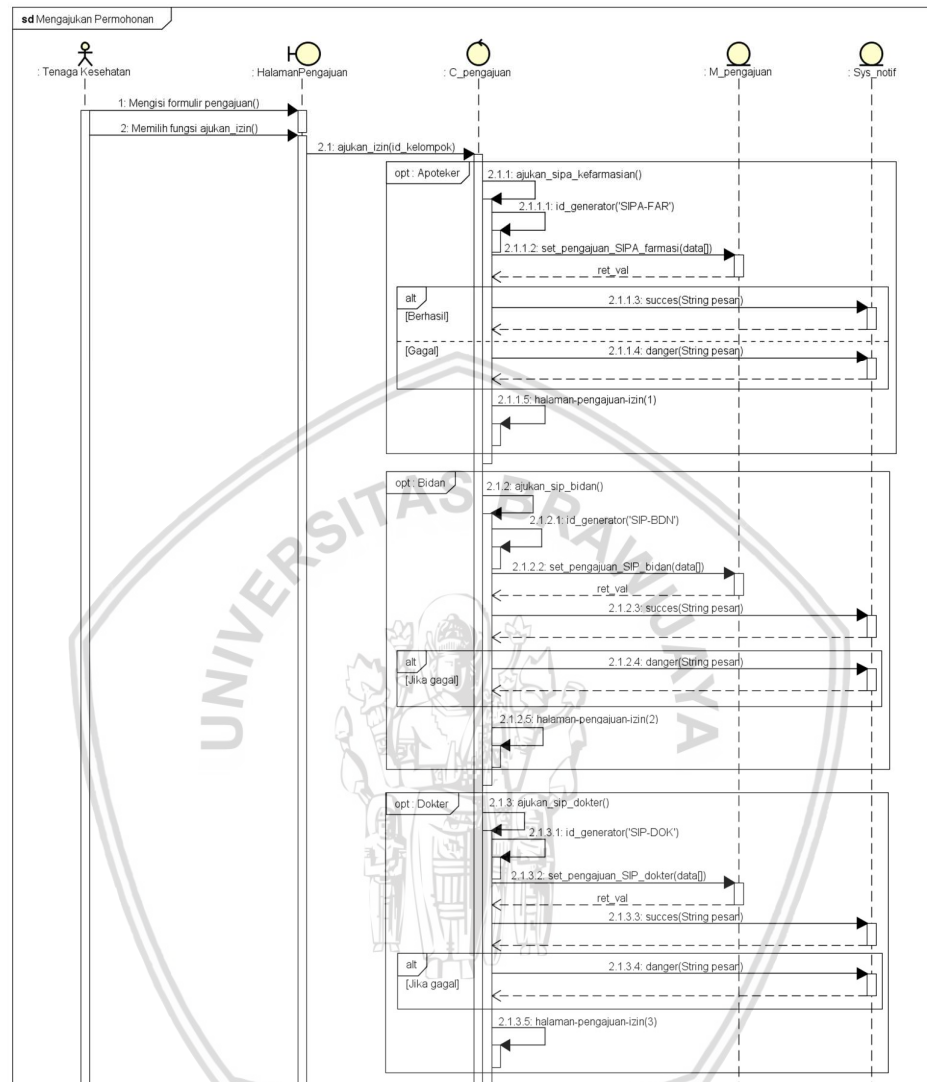
5.2 Sequence diagram

Pada bagian ini, alur proses yang terjadi di dalam SINKES digambarkan berdasarkan urutan waktu. *Sequence diagram* yang dimodelkan pada bagian ini merupakan alur proses dari fungsional utama pada SINKES, sedangkan *class diagram* yang akan digambarkan pada poin 5.3 merupakan kumpulan kelas, atribut dan fungsi dari seluruh fungsional yang telah didefinisikan pada Bab 4.

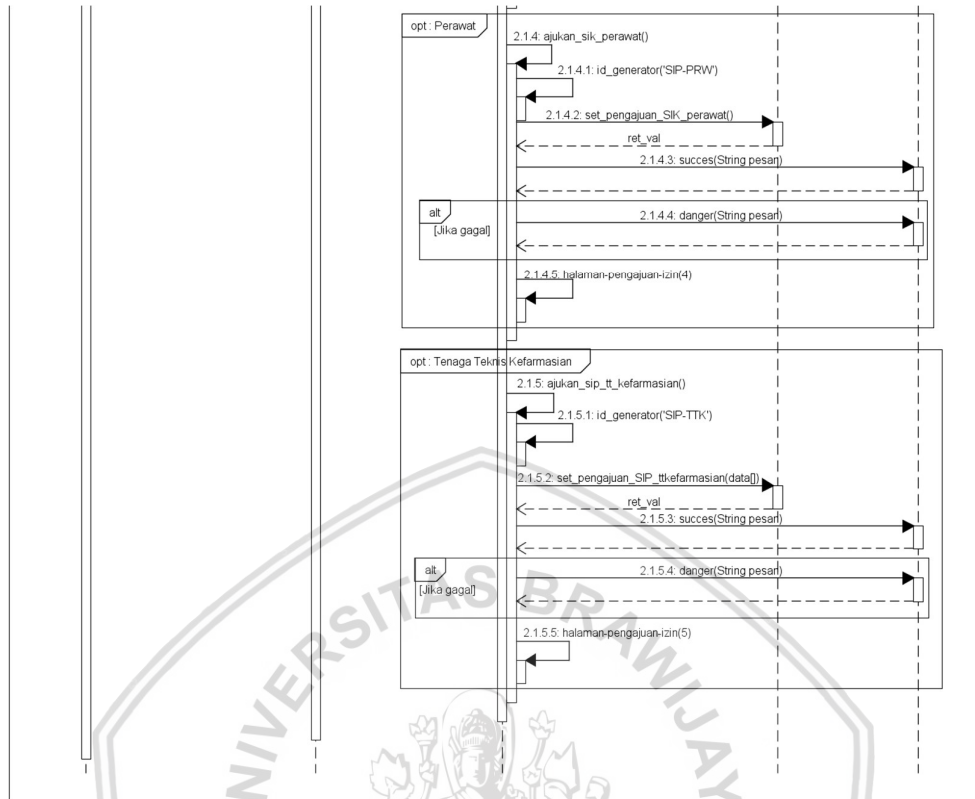
5.2.1 Sequence diagram mengajukan permohonan

Sequence diagram mengajukan permohonan yang tunjukkan dalam Gambar 5.2 dan 5.3 menunjukkan alur proses ketika Tenaga Kesehatan mengajukan permohonan surat izin yang mengacu pada *use case* dengan kode UCT-05. Proses mengajukan permohonan ini melibatkan 3 kelas dan 1 *interface*. *Interface* terdiri dari *HalamanPengajuan* yang berisi formulir pengisian data dan berkas persyaratan. Sedangkan pada sisi *controller*, fungsi-fungsi yang menangani

pengajuan berada pada kelas C_pengajuan. Kemudian pada sisi *model*, terdapat kelas M_pengajuan dan Sys_notif.



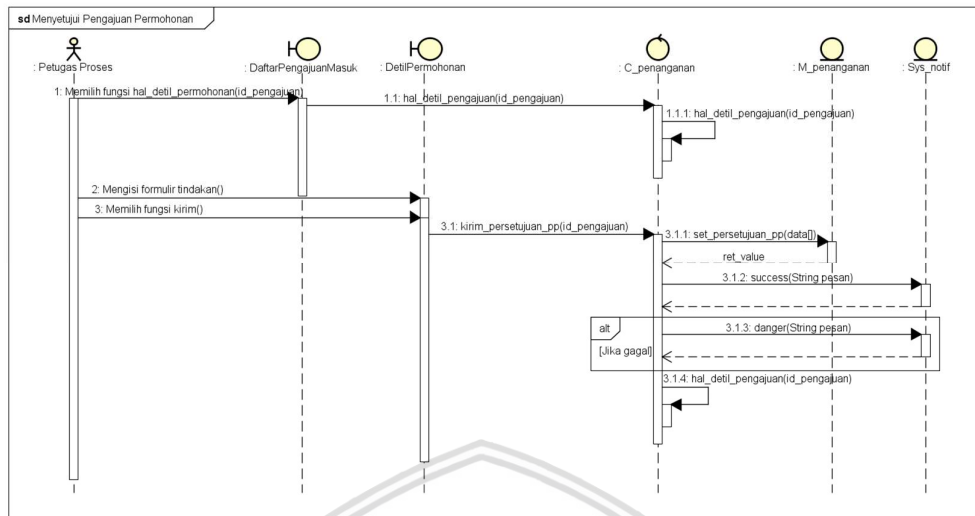
Gambar 5.2 *Sequence diagram* mengajukan permohonan



Gambar 5.2 *Sequence diagram* mengajukan permohonan (lanjutan)

5.2.2 *Sequence diagram* menyetujui pengajuan permohonan

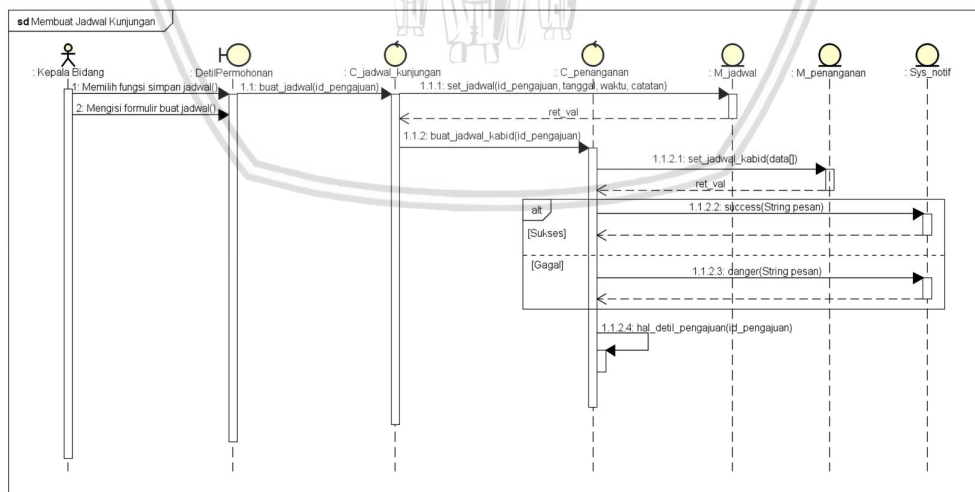
Sequence diagram menyetujui pengajuan permohonan yang tunjukkan dalam Gambar 5.4 menunjukkan alur proses ketika Petugas Proses melakukan tindakan menyetujui permohonan yang telah diajukan oleh Tenaga Kesehatan yang mengacu pada *use case* dengan kode UCP-04. Proses menyetujui pengajuan permohonan ini melibatkan 3 kelas dan 2 *interface*. *Interface* terdiri dari DaftarPengajuanMasuk yang berisi daftar permohonan yang masuk di SINKES dan DetilPermohonan yang berisi data pemohon, berkas persyaratan yang dilampirkan, dan formulir tindakan yang akan diambil. Sedangkan pada sisi *controller*, fungsi-fungsi yang menangani penanganan berada pada kelas C_penanganan. Kemudian pada sisi *model*, terdapat kelas M_penanganan dan Sys_notif.



Gambar 5.3 Sequence diagram menyetujui berkas permohonan

5.2.3 Sequence diagram membuat jadwal kunjungan

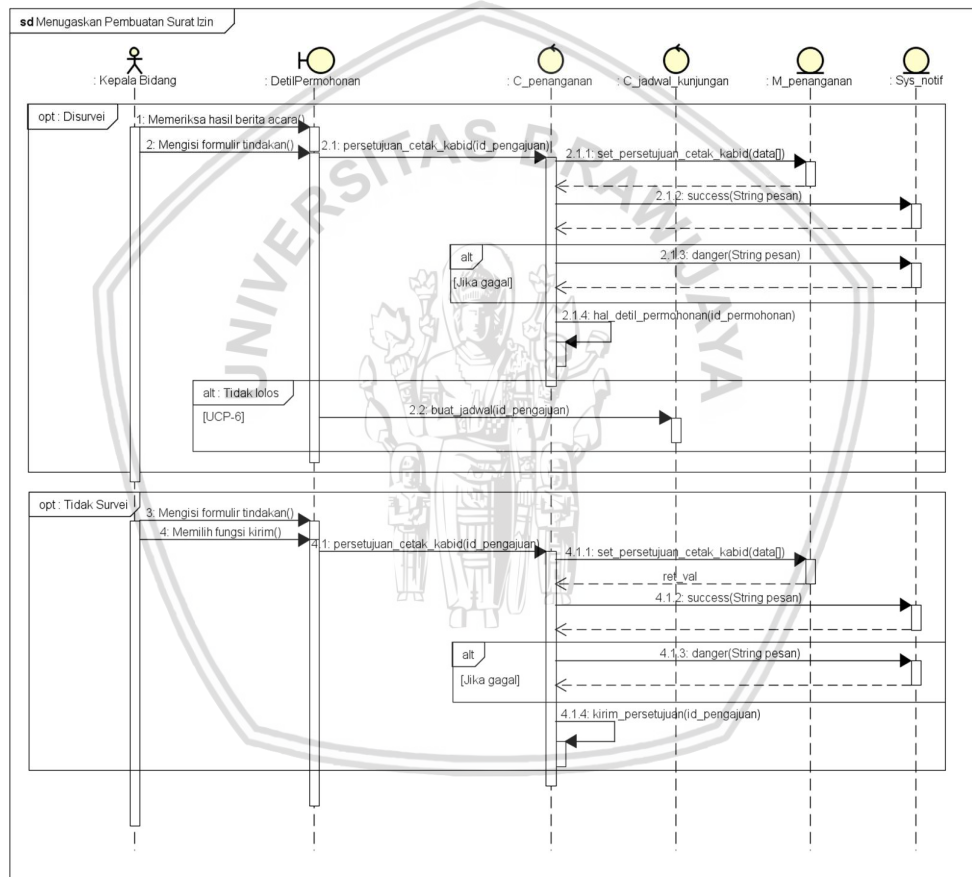
Sequence diagram membuat jadwal kunjungan yang tunjukkan dalam Gambar 5.5 menunjukkan alur proses ketika Kepala Bidang melakukan penjadwalan kunjungan survei yang mengacu pada *use case* dengan kode UCP-06. Proses membuat jadwal kunjungan ini melibatkan 5 kelas dan 1 *interface*. *Interface* terdiri dari *DetilPengajuan* yang berisi data pemohon, berkas persyaratan yang dilampirkan, dan formulir pembuatan jadwal. Sedangkan pada sisi *controller*, fungsi-fungsi yang menangani pembuatan jadwal berada pada kelas *C_jadwal_kunjungan* dan *C_penanganan*. Kemudian pada sisi *model*, terdapat kelas *M_jadwal*, *M_penanganan* dan *Sys_notif*.



Gambar 5.4 Sequence diagram menambahkan jadwal kunjungan

5.2.4 Sequence diagram menugaskan pembuatan surat tugas

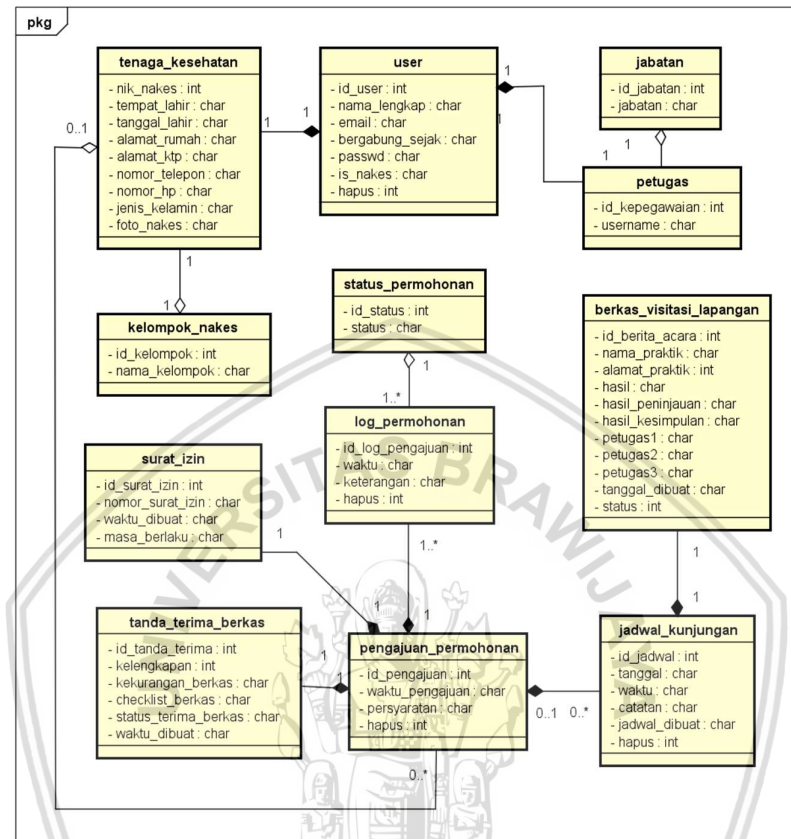
Sequence diagram menugaskan pembuatan surat tugas yang tunjukkan dalam Gambar 5.7 menunjukkan alur proses sistem ketika Kepala Bidang menugaskan pembuatan surat tugas kepada Petugas Proses setelah seluruh proses penanganan telah dilewati dan hasil survei bagi Tenaga Kesehatan yang disurvei dinyatakan lolos. *Sequence diagram* tersebut mengacu pada *use case* dengan kode UCP-08. Proses membuat berita acara ini melibatkan 4 kelas dan 1 *interface*. *Interface* terdiri dari DetilPermohonan yang berisi data pemohon, berkas persyaratan yang dilampirkan, dan formulir penugasan. Sedangkan pada sisi *controller*, terdapat kelas C_penanganan. Kemudian pada sisi *model*, terdapat kelas M_penanganan dan Sys_notif.



Gambar 5.5 Sequence diagram menugaskan pembuatan surat tugas

5.3 Perancangan *class diagram*

5.3.1 Diagram kelas analisis



Gambar 5.6 Diagram kelas analisis

Diagram kelas analisis yang ditunjukkan dalam gambar 5.6 berisi visualisasi hubungan antar kelas sebagai representasi objek yang menjadi komponen penyusun sistem yang akan dikembangkan. Kelas-kelas di diagram didapatkan berdasarkan pemilihan kosakata yang diidentifikasi berdasarkan hasil analisis yang telah dilakukan. Dalam diagram tersebut terdapat 12 kelas dengan masing-masing kelas memiliki atribut-atribut tertentu. Masing-masing kelas memiliki hubungan dengan kelas yang lain. Diagram kelas analisis tersebut akan digunakan sebagai panduan pemodelan kelas perancangan dan perancangan basis data.

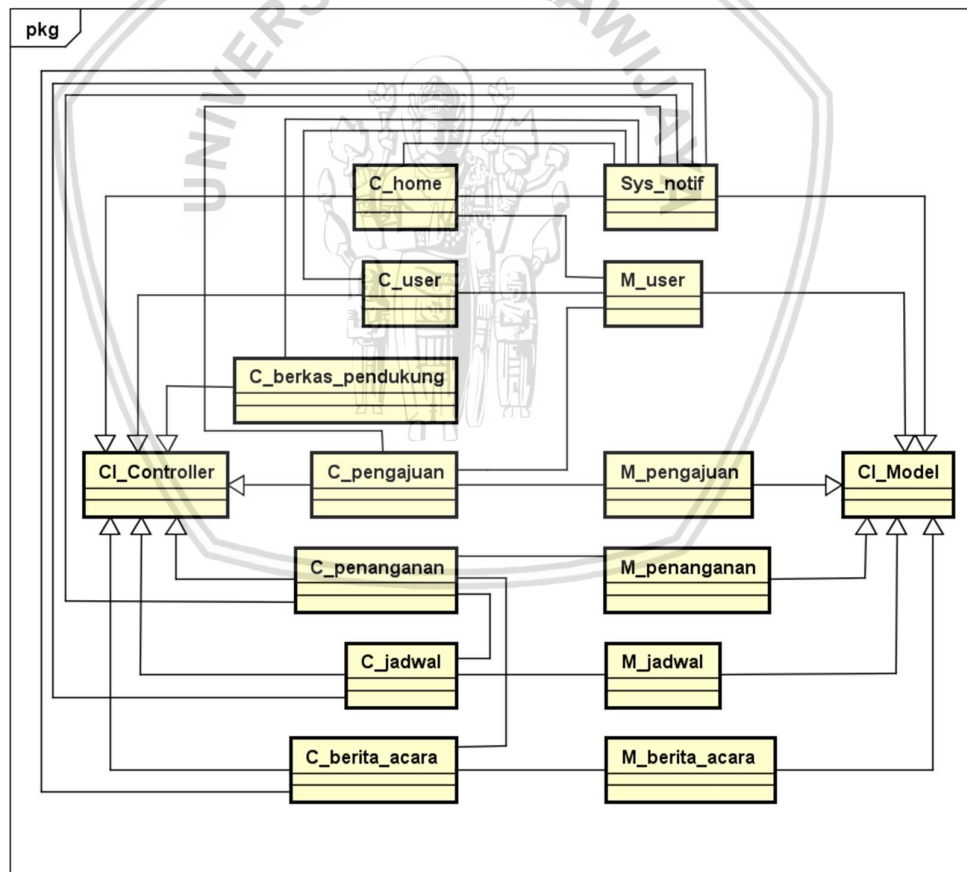
Pemodelan objek yang divisualisasikan dalam Gambar 5.6 menunjukkan kelas User merupakan representasi objek pengguna yang memuat beberapa atribut. Kelas ini terhubung dengan kelas tenaga_kesehatan dan petugas, yang masing-masing kelas tersebut memiliki relasi komposisi yang berarti memiliki ketergantungan kepada objek User yang berisi informasi dasar yang dimiliki oleh pengguna sistem. Kelas kelompok_nakes memiliki relasi dengan kelas tenaga_kesehatan dikarenakan setiap tenaga_kesehatan pasti memiliki jenis atau kelompok profesinya seperti Apoteker atau Dokter. Sedangkan kelas jabatan memiliki relasi dengan kelas petugas sebagai perwujudan dari objek jabatan yang

harus dimiliki oleh tiap petugas agar dapat menjalankan tugasnya sesuai dengan otoritas yang dimiliki.

Selanjutnya objek pengajuan permohonan divisualisasikan dengan kelas pengajuan_permohonan yang berisi informasi terkait permohonan yang diajukan oleh Tenaga Kesehatan. Adapun kelas-kelas tanda_terima_berkas, jadwal_kunjungan, berkas_visitasi_lapangan, log_permohonan dan surat_izin merupakan visualisasi dari objek yang terbentuk setelah objek pengajuan_permohonan terbentuk.

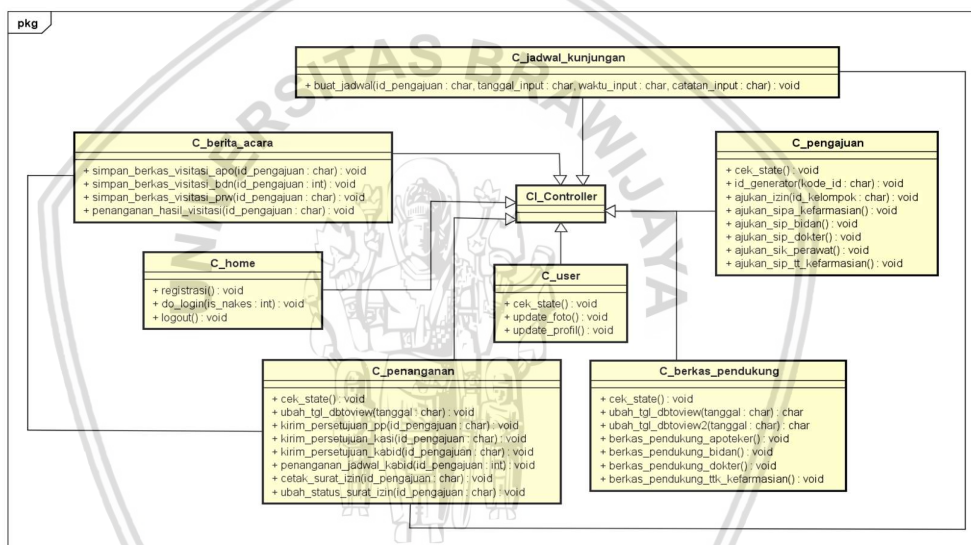
5.3.2 Diagram kelas perancangan

Class diagram yang digambarkan pada bagian ini dijadikan sebagai acuan dari kelas-kelas yang akan digunakan dalam pembangunan *website SINKES*. *Class diagram* digambarkan menjadi 3 bagian, yaitu antara lain *class diagram* yang menggambarkan relasi antar kelas secara keseluruhan, *class diagram* yang menggambarkan relasi kelas-kelas *controller* dan *class diagram* yang menggambarkan relasi kelas-kelas *model* pada *website SINKES*.



Gambar 5.7 *class diagram* relasi antar kelas SINKES

Gambar 5.6 menunjukkan hubungan atau relasi yang dimiliki kelas *logical* dan model data. Penamaan kelas *logical* diawali huruf “C”, sedangkan kelas model data menggunakan huruf “M” dengan maksud untuk mempermudah identifikasi jenis kelas. Masing-masing kelas *logical* terhubung secara generalisasi dengan kelas *CI_Controller* yang telah tersedia di kerangka kerja *Codeigniter*. Beberapa kelas *logical* seperti *C_penanganan*, *C_jadwal* dan *C_berita* terhubung satu sama lain untuk menunjukkan bahwa kelas-kelas tersebut saling terkait pada pemanggilan *method* yang dimiliki. Kemudian pada kelas model data, masing-masing juga terhubung secara generalisasi dengan kelas *CI_Model* yang telah disediakan kerangka kerja. Dari semua kelas yang terhubung pada kelas *CI_Model* tersebut, terdapat sebuah kelas bernama *Sys_notif* yang tidak memiliki hubungan langsung dengan proses CRUD (*Create, Read, Update, Delete*) pada *database*, melainkan hanya sebagai kelas yang memiliki fungsi untuk melemparkan nilai berupa *tag HTML alert*.



Gambar 5.8 Class diagram controller SINKES

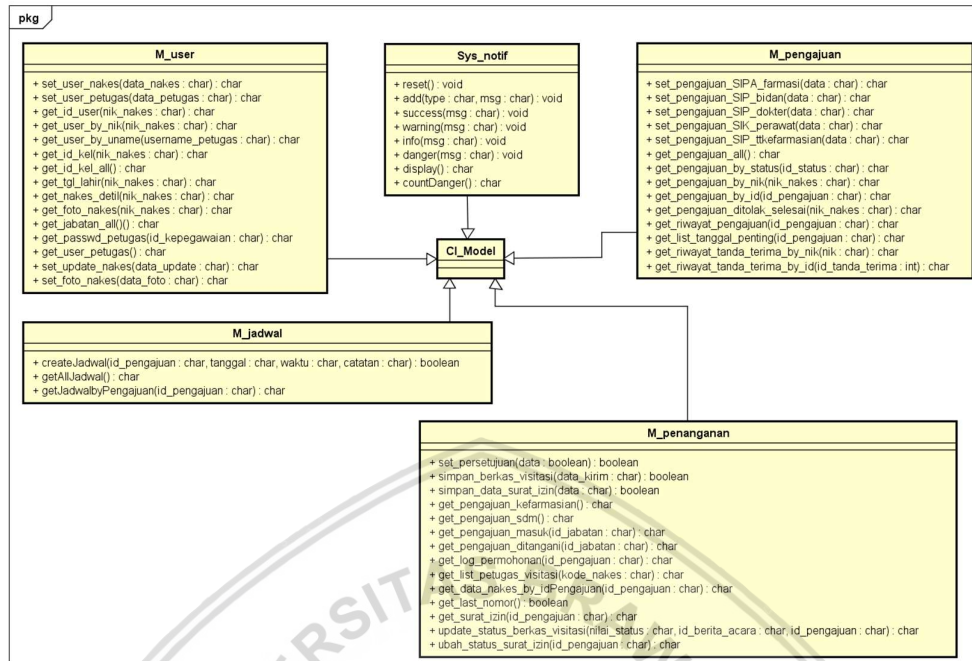
Selanjutnya pada Gambar 5.7 yang menunjukkan kelas-kelas *logical* SINKES. Terdapat 8 kelas yang bernama *CI_Controller*, *C_jadwal_kunjungan*, *C_berita_acara*, *C_pengajuan*, *C_home*, *C_user*, *C_penanganan*, *C_berkas_pendukung*. Berikut pada Tabel 5.1 merupakan penjelasan isi masing-masing kelas.

Tabel 5.1 Penjelasan kelas-kelas pada kelas *logical*

No.	Nama Kelas	Penjelasan
1.	<i>CI_Controller</i>	Kelas yang telah disediakan kerangka kerja Codeigniter. Di ekstensi oleh kelas lainnya agar dapat menggunakan fungsi-fungsi yang telah disediakan
2.	<i>C_jadwal_kunjungan</i>	Kelas yang berisi fungsi-fungsi untuk menampilkan halaman dan mengolah masukan pengguna terkait

Tabel 5.1 Penjelasan kelas-kelas pada kelas *logical*

No.	Nama Kelas	Penjelasan
		pembuatan jadwal oleh Kepala Bidang dan melihat daftar jadwal yang telah dibuat.
3.	C_berita_acara	Kelas yang berisi fungsi-fungsi untuk menampilkan halaman dan mengolah masukan pengguna terkait pembuatan berita acara setelah kunjungan survei dilakukan.
4.	C_pengajuan	Kelas yang berisi fungsi-fungsi untuk menampilkan halaman dan mengolah masukan pengguna berjenis Tenaga Kesehatan terkait aktivitas mengajukan permohonan surat izin kerja/praktik.
5.	C_home	Kelas yang berisi fungsi-fungsi untuk menampilkan halaman dan mengolah logik untuk kebutuhan registrasi, <i>login</i> dan <i>logout</i> .
6.	C_user	Kelas yang berisi fungsi-fungsi untuk menampilkan halaman dan mengolah masukan pengguna berjenis Tenaga Kesehatan dalam aktivitas yang berhubungan dengan pengolahan profil pengguna.
7.	C_penanganan	Kelas yang berisi fungsi-fungsi untuk menampilkan halaman dan mengolah masukan pengguna terkait aktivitas yang berhubungan dengan penanganan pihak Dinkes terhadap permohonan surat izin yang telah diajukan oleh Tenaga Kesehatan.
8.	C_berkas_pendukung	Kelas yang berisi fungsi-fungsi untuk menampilkan halaman dan mengolah masukan pengguna berjenis Tenaga Kesehatan terkait aktivitas membuat berkas pendukung yang <i>template</i> -nya telah disediakan oleh pihak Dinkes.



Gambar 5.9 Class diagram model website SINKES

Kemudian pada Gambar 5.8 yang menunjukkan kelas-kelas model data SINKES. Terdapat 6 kelas yang bernama CI_Model, M_jadwal, M_penanganan, M_pengajuan, M_user, Sys_notif. Berikut pada Tabel 5.2 merupakan penjelasan isi masing-masing kelas.

Tabel 5.2 Penjelasan kelas-kelas pada kelas model

No.	Nama Kelas	Penjelasan
1.	CI_Model	Kelas yang telah disediakan kerangka kerja Codeigniter. Di ekstensi oleh kelas lainnya agar dapat menggunakan fungsi-fungsi yang telah disediakan
2.	M_jadwal	Kelas yang berisi fungsi-fungsi untuk mengolah <i>query database</i> terkait aktivitas membuat dan melihat daftar jadwal kunjungan kerja.
3.	M_penanganan	Kelas yang berisi fungsi-fungsi untuk mengolah <i>query database</i> terkait aktivitas penanganan permohonan, khususnya pencatatan riwayat penanganan yang telah dilakukan oleh pihak Dinkes.
4.	M_pengajuan	Kelas yang berisi fungsi-fungsi untuk mengolah <i>query database</i> terkait aktivitas pengajuan permohonan yang diajukan oleh Tenaga Kesehatan.
5.	M_user	Kelas yang berisi fungsi-fungsi untuk mengolah <i>query database</i> terkait aktivitas yang berhubungan dengan pengguna, seperti pembuatan akun, merubah profil, hingga pencarian pengguna saat proses otentifikasi.

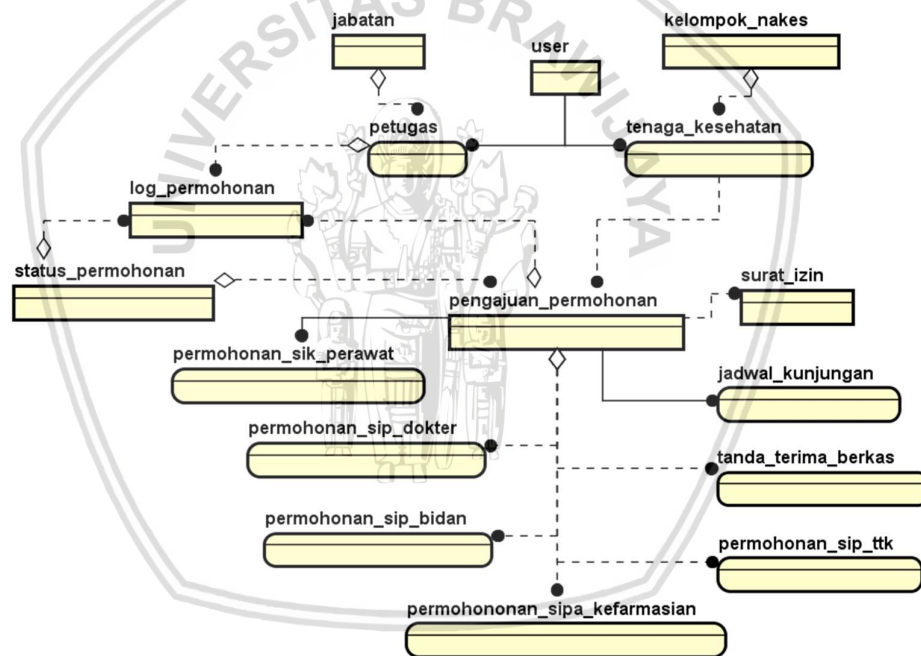
Tabel 5.2 Penjelasan kelas-kelas pada kelas model

No.	Nama Kelas	Penjelasan
6.	Sys_notif	Kelas yang berisi fungsi-fungsi untuk mengirimkan <i>tag alert</i> HTML ke view.

5.4 Perancangan basisdata

5.4.1 Perancangan *Conceptual Data Model* (CDM)

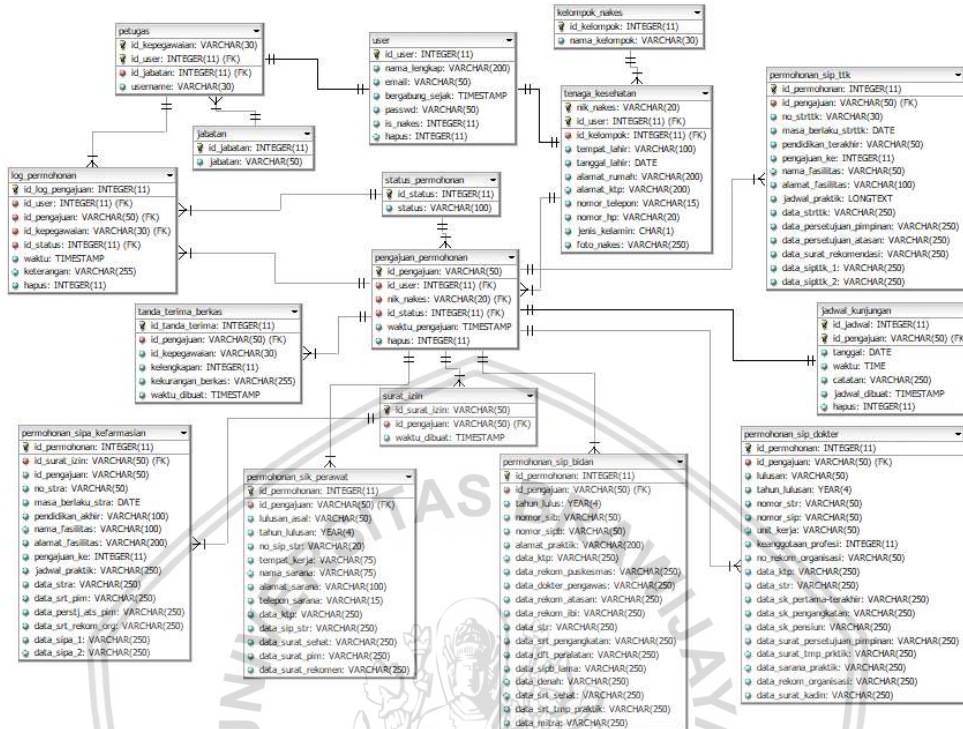
Seperti yang dijelaskan pada Bab 2 poin 2.7, perancangan *Conceptual Data Model* (CDM) berguna untuk menggambarkan entitas dalam bentuk tabel-tabel tempat penyimpanan data dan relasi antar tabel tersebut. Perancangan CDM pada pembangunan *website* SINKES yang akan dijadikan dasar untuk membuat perancangan *Physical Data Model* (PDM) ditunjukkan dalam Gambar 5.10 berikut.

Gambar 5.10 Perancangan *Conceptual Data Model* (CDM)

5.4.2 Perancangan *Physical Data Model* (PDM)

Bagian ini merupakan penggambaran skema basis data yang digunakan sebagai acuan untuk membangun basis data dari SINKES. Perancangan PDM dibuat dengan menggunakan *tool DBDesigner* yang dapat digunakan untuk memodelkan perancangan basisdata dengan jelas. Setiap tabel digambarkan dengan bentuk kotak, sedangkan relasi antar tabelnya digambarkan dengan garis. Pada kotak Tabel, logo kunci menggambarkan *primary key* (PK), belah ketupat merah menggambarkan *foreign key* (FK) dan belah ketupat berwarna biru

menggambarkan kolom yang dimiliki tabel. Perancangan PDM SINKES ditunjukkan dalam Gambar 5.11.



Gambar 5.11 Perancangan *Physical Data Model (PDM)*

5.5 Perancangan antarmuka

Pada bagian ini akan ditunjukkan perancangan antarmuka pengguna yang dilakukan untuk menggambarkan tampilan dari sistem yang akan dibangun.

5.5.1 Perancangan antarmuka mengajukan surat izin

Berikut pada Gambar 5.12 merupakan rancangan antarmuka Tenaga Kesehatan untuk halaman pengajuan permohonan. Keterangan dari Gambar 5.12 antara lain adalah.

1. *Sidebar* (Menu navigasi)
2. *Callout* (Peringat)
3. Formulir Permohonan
4. Tombol Pengajuan

The screenshot shows a web application for health staff to apply for a Practical License (SIP). The interface is divided into several sections:

- Sidebar (1):** A vertical menu on the left containing links for 'Daftar Menu', 'Beranda', 'Kelengkapan Data Diri >', 'Profil', 'Berkas Pendukung', and 'Pengajuan Surat Izin'.
- Header (2):** A top bar with the title 'Permohonan Surat Izin Praktik (SIP)' and a breadcrumb trail 'Pengajuan Surat Izin > Permohonan'.
- Reminder (3):** A large orange box with the text 'Pesan pengingat' and a sub-header 'Pastikan Anda telah memenuhi seluruh berkas persyaratan yang ditentukan!'.
- Form (4):** A large white box containing a form for application details, including fields for 'Nama', 'Jenis Tenaga Kesehatan', and 'Ajukan Permohonan'.

Gambar 5.12 Perancangan antarmuka mengajukan surat izin

5.5.2 Perancangan antarmuka menyetujui pengajuan permohonan

Berikut pada Gambar 5.13 merupakan rancangan antarmuka Petugas Proses untuk halaman ketika melakukan persetujuan terhadap permohonan yang diajukan oleh Tenaga Kesehatan. Keterangan dari Gambar 5.13 antara lain adalah.

1. *Sidebar* (Menu navigasi)
2. *Callout* (Informasi)
3. Tabel berisi informasi log penanganan permohonan
4. Formulir pengisian tindakan
5. Tombol mengirimkan tindakan
6. Data Pemohon

The screenshot shows the 'Admin SINKES' interface. On the left is a sidebar (1) with a login form (Name, Online, Logout) and a menu (Dashboard, Berkas Pemohon, Semua Pemohonan, Sedang ditangani, Jadwal Survei, Akun, Password). The main content area is titled 'Detail Pengajuan' and includes a breadcrumb 'Pengajuan Surat Izin > Permohonan'. It contains three sections: 'Informasi' (2) with a table (3) of columns Tanggal, Waktu, Petugas, and Status; 'Tindakan' (4) with input fields and a 'Kirim' button (5); and 'Data Pemohonan' (6) with a table.

Gambar 5.13 Perancangan antarmuka menyetujui pengajuan permohonan

5.5.3 Perancangan antarmuka membuat jadwal kunjungan

Berikut pada Gambar 5.14 merupakan rancangan antarmuka Kepala Bidang untuk halaman ketika membuat jadwal kunjungan untuk melakukan survei kelayakan tempat praktik Tenaga Kesehatan yang mengajukan permohonan surat izin. Keterangan dari Gambar 5.14 antara lain adalah.

1. *Sidebar* (Menu navigasi)
2. *Callout* (Informasi)
3. Tabel berisi informasi log penanganan permohonan
4. Formulir jadwal kunjungan
5. Tombol mengirimkan jadwal
6. Data Pemohon

The screenshot shows the 'Admin SINKES' interface. On the left is a sidebar (1) with a user profile section (Name, Online, Logout) and a 'Menu Navigasi' section containing links: Dashboard, Berkas Pemohon >, Semua Permohonan, Sedang ditangani, Jadwal Survei, Akun, and Password. The main content area is titled 'Detil Pengajuan' and includes a breadcrumb 'Pengajuan Surat Izin > Permohonan'. It contains an 'Informasi' section (2), a table with columns 'Tanggal', 'Waktu', 'Petugas', and 'Status' (3), a 'Buat Jadwal' form (4) with input fields and a 'Kirim' button (5), and a 'Data Permohonan' table (6). A large watermark of Universitas Brawijaya is visible in the background.

Gambar 5.14 Perancangan antarmuka membuat jadwal kunjungan

5.5.4 Perancangan antarmuka menugaskan pembuatan surat tugas

Berikut pada Gambar 5.15 merupakan rancangan antarmuka Kepala Bidang untuk halaman ketika melakukan penugasan pencetakan surat izin. Halaman ini dapat diakses ketika permohonan yang diajukan telah disetujui Petugas Proses dan Kepala Seksi. Keterangan dari Gambar 5.15 antara lain adalah.

1. *Sidebar* (Menu navigasi)
2. *Callout* (Informasi)
3. Tabel berisi informasi log penanganan permohonan
4. Formulir pengisian tindakan
5. Tombol mengirimkan tindakan
6. Data Pemohon

The screenshot displays the 'Admin SINKES' web application. On the left is a sidebar menu (1) with options like 'Logout', 'Menu Navigasi', 'Dashboard', 'Berkas Pemohon >', 'Semua Permohonan', 'Sedang ditangani', 'Jadwal Survei', 'Akun', and 'Password'. The main content area is titled 'Detail Pengajuan' (2) and shows a table (3) with columns 'Tanggal', 'Waktu', 'Petugas', and 'Status'. Below this is a 'Penugasan Pencetakan Surat Izin' section (4) containing input fields and a green 'Kirim' button (5). At the bottom is a 'Data Permohonan' table (6) with multiple rows for data entry.

Gambar 5.15 Perancangan antarmuka menugaskan pembuatan surat tugas

5.6 Perancangan algoritme

Pada bagian perancangan algoritme berikut, dijelaskan mengenai algoritme yang akan diterapkan untuk implementasi SINKES.

5.6.1 Algoritme mengajukan permohonan

Algoritme *function* *ajukan_izin*(\$id_kelompok) merupakan gambaran fungsi sistem untuk mengirimkan formulir permohonan izin yang telah diisi oleh Tenaga Kesehatan. *Function* *ajukan_izin*(\$id_kelompok) adalah bagian dari *controller* *C_pengajuan*. Algoritme dimulai dengan memasukkan parameter *id_kelompok* pada *function* *ajukan_izin*. Kemudian akan dilakukan pemeriksaan sebagai dasar pemanggilan fungsi yang sesuai dengan Tenaga Kesehatan yang sedang mengirimkan permohonan.

Tabel 5.3 Algoritme mengajukan permohonan

No	Pseudocode
1.	Mulai
2.	Periksa <i>id_kelompok</i>
3.	Jika <i>id_kelompok</i> bernilai 1, maka jalankan fungsi
4.	mengajukan izin untuk Apoteker
5.	Berhenti
6.	Jika <i>id_kelompok</i> bernilai 2, maka jalankan fungsi
7.	mengajukan izin untuk Bidan

Tabel 5.3 Algoritme mengajukan permohonan

No	Pseudocode
8.	Berhenti
9.	Jika id_kelompok bernilai 3, maka jalankan fungsi
10.	mengajukan izin untuk Dokter
11.	Berhenti
12.	Jika id_kelompok bernilai 4, maka jalankan fungsi
13.	mengajukan izin untuk Perawat
14.	Selesai
15.	Jika id_kelompok bernilai 5, maka jalankan fungsi
16.	mengajukan izin untuk Tenaga Teknis Kefarmasian
17.	Selesai
18.	

Berikut pada Tabel 5.2 adalah algoritme yang dijalankan jika Tenaga Kesehatan yang akan melakukan pengajuan teridentifikasi sebagai kelompok Apoteker.

Tabel 5.4 Algoritme mengajukan permohonan untuk Apoteker

No	Pseudocode
1.	Mulai
2.	Simpan nomor stra ke variabel lokal
3.	Simpan masa berlaku stra ke variabel lokal
4.	Simpan pendidikan akhir ke variabel lokal
5.	Simpan permohonan surat ke variabel lokal
6.	Simpan nama fasilitas ke variabel lokal
7.	Simpan alamat fasilitas ke variabel lokal
8.	Simpan jadwal praktik ke variabel lokal
9.	Panggil fungsi id_generator
10.	
11.	Atur konfigurasi tipe file upload
12.	Atur konfigurasi ukuran maksimal file upload
13.	
14.	For i=1; i<=6; 1++
15.	Switch i
16.	Jika i = 1, maka simpan letak penyimpanan berkas stra
17.	ke variabel lokal
18.	Jika i = 2, maka simpan letak penyimpanan berkas
19.	pernyataan memiliki tempat praktik ke variabel lokal
20.	Jika i = 3, maka simpan letak penyimpanan berkas surat
21.	pernyataan atasan ke variabel lokal
22.	Jika i = 4, maka simpan letak penyimpanan berkas sipa
23.	1 ke variabel lokal
24.	Jika i = 5, maka simpan letak penyimpanan berkas sipa
25.	1 ke variabel lokal
26.	Jika i = 6, maka simpan letak penyimpanan berkas sipa
27.	1 ke variabel lokal
28.	End switch
29.	
30.	Simpan nama file upload ke variabel lokal
31.	Simpan nama baru file ke variabel lokal
32.	Atur konfigurasi nama file dengan nama baru
33.	Atur konfigurasi letak penyimpanan dengan nilai variabel
34.	lokal yang menyimpan letak penyimpanan file
35.	Inisialisasi konfigurasi

Tabel 5.4 Algoritme mengajukan permohonan untuk Apoteker

No	Pseudocode
36.	
37.	IF file upload tidak kosong
38.	Simpan data file upload ke variabel array
39.	Simpan letak penyimpanan file ke variabel array
40.	ELSE isikan null ke variabel array penyimpan letak file
41.	ENDIF
42.	
43.	Increment nilai i
44.	END FOR
45.	
46.	Simpan nilai variabel yang menyimpan nilai input user ke
47.	variabel array
48.	
49.	Panggil fungsi untuk menyimpan data ke model dan simpan
50.	nilai kembalian ke variabel
51.	
52.	IF nilai kembalian 'success'
53.	Panggil fungsi alert sukses
54.	ELSE panggil fungsi alert gagal
55.	ENDIF
56.	
57.	Arahkan ke halaman detail pengajuan
58.	
59.	Selesai

Berikut pada Tabel 5.3 adalah algoritme yang dijalankan jika Tenaga Kesehatan yang akan melakukan pengajuan teridentifikasi sebagai kelompok Bidan.

Tabel 5.5 Algoritme mengajukan permohonan untuk Bidan

No	Pseudocode
1.	Mulai
2.	Simpan tahun lulus ke varabel lokal
3.	Simpan nomor sib ke varabel lokal
4.	Simpan nomor sipb ke varabel lokal
5.	Simpan alamat praktik ke varabel lokal
6.	Panggil fungsi id_generator
7.	
8.	Atur konfigurasi tipe file upload
9.	Atur konfigurasi ukuran maksimal file upload
10.	
11.	For i=1; i<=6; 1++
12.	Switch i
13.	Jika i = 1, maka simpan letak penyimpanan berkas ktp
14.	ke variabel lokal
15.	Jika i = 2, maka simpan letak penyimpanan berkas
16.	Rekomendasi Puskesmas Wilayah ke variabel lokal
17.	Jika i = 3, maka simpan letak penyimpanan berkas Surat
18.	Kesanggupan Dokter Pengawas ke variabel lokal
19.	Jika i = 4, maka simpan letak penyimpanan berkas Surat
20.	Rekomendasi Atasan ke varabel lokal
21.	Jika i = 5, maka simpan letak penyimpanan berkas
22.	Rekomendasi IBI ke varabel lokal

Tabel 5.5 Algoritme mengajukan permohonan untuk Bidan

No	Pseudocode
23.	Jika i = 6, maka simpan letak penyimpanan berkas STR
24.	berlaku dan terlegalisir ke variabel lokal
25.	Jika i = 7, maka simpan letak penyimpanan berkas SK
26.	Pengangkatan Penempatan ke variabel lokal
27.	Jika i = 8, maka simpan letak penyimpanan berkas Daftar
28.	Peralatan ke variabel lokal
29.	Jika i = 9, maka simpan letak penyimpanan berkas SIPB
30.	Lama ke variabel lokal
31.	Jika i = 10, maka simpan letak penyimpanan berkas
32.	Gambar Denah Bangunan ke variabel lokal
33.	Jika i = 11, maka simpan letak penyimpanan berkas Surat
34.	Sehat ke variabel lokal
35.	Jika i = 12, maka simpan letak penyimpanan berkas Surat
36.	Pernyataan Kepemilikan ke variabel lokal
37.	Jika i = 13, maka simpan letak penyimpanan berkas Surat
38.	Bidan Mitra ke variabel lokal
39.	
40.	End switch
41.	
42.	Simpan nama file upload ke variabel lokal
43.	Simpan nama baru file ke variabel lokal
44.	Atur konfigurasi nama file dengan nama baru
45.	Atur konfigurasi letak penyimpanan dengan nilai variabel
46.	lokal yang menyimpan letak penyimpanan file
47.	Inisialisasi konfigurasi
48.	
49.	IF file upload tidak kosong
50.	Simpan data file upload ke variabel array
51.	Simpan letak penyimpanan file ke variabel array
52.	ELSE isikan null ke variabel array penyimpan letak file
53.	ENDIF
54.	
55.	Increment nilai i
56.	END FOR
57.	
58.	Simpan nilai variabel yang menyimpan nilai input user ke
59.	variabel array
60.	
61.	Panggil fungsi untuk menyimpan data ke model dan simpan
62.	nilai kembalian ke variabel
63.	
64.	IF nilai kembalian 'success'
65.	Panggil fungsi alert sukses
66.	ELSE panggil fungsi alert gagal
67.	ENDIF
68.	
69.	Arahkan ke halaman detil pengajuan
70.	
71.	Selesai
72.	

Berikut pada Tabel 5.4 adalah algoritme yang dijalankan jika Tenaga Kesehatan yang akan melakukan pengajuan teridentifikasi sebagai kelompok Dokter.

Tabel 5.6 Algoritme mengajukan permohonan untuk Dokter

No	Pseudocode
1.	Mulai
2.	Simpan asal lulusan ke variabel lokal
3.	Simpan tahun lulus ke variabel lokal
4.	Simpan nomor str ke variabel lokal
5.	Simpan nomor sip ke variabel lokal
6.	Simpan unit kerja ke variabel lokal
7.	Simpan kenggotaan idi ke variabel lokal
8.	Simpan nomor rekomendasi ke variabel lokal
9.	Panggil fungsi id_generator
10.	
11.	Atur konfigurasi tipe file upload
12.	Atur konfigurasi ukuran maksimal file upload
13.	
14.	For i=1; i<=6; 1++
15.	Switch i
16.	Jika i = 1, maka simpan letak penyimpanan berkas ktp
17.	ke variabel lokal
18.	Jika i = 2, maka simpan letak penyimpanan berkas STR
19.	ke variabel lokal
20.	Jika i = 3, maka simpan letak penyimpanan berkas SK
21.	Pangkatangan Awal-Akhir ke variabel lokal
22.	Jika i = 4, maka simpan letak penyimpanan berkas SK
23.	Pegawai ke variabel lokal
24.	Jika i = 5, maka simpan letak penyimpanan berkas SK
25.	pensiun ke variabel lokal
26.	Jika i = 6, maka simpan letak penyimpanan berkas Surat
27.	Persetujuan Atasan ke variabel lokal
28.	Jika i = 7, maka simpan letak penyimpanan berkas Surat
29.	Pernyataan Memiliki Tempat Praktik ke variabel lokal
30.	Jika i = 8, maka simpan letak penyimpanan berkas Surat
31.	Persetujuan dari Tempat Praktik ke variabel lokal
32.	Jika i = 9, maka simpan letak penyimpanan berkas Surat
33.	Rekomendasi dari IDI ke variabel lokal
34.	Jika i = 10, maka simpan letak penyimpanan berkas Surat
35.	Persetujuan Kepala Dinas ke variabel lokal
36.	
37.	End switch
38.	
39.	Simpan nama file upload ke variabel lokal
40.	Simpan nama baru file ke variabel lokal
41.	Atur konfigurasi nama file dengan nama baru
42.	Atur konfigurasi letak penyimpanan dengan nilai variabel
43.	lokal yang menyimpan letak penyimpanan file
44.	Inisialisasi konfigurasi
45.	
46.	IF file upload tidak kosong
47.	Simpan data file upload ke variabel array
48.	Simpan letak penyimpanan file ke variabel array
49.	ELSE isikan null ke variabel array penyimpan letak file
50.	ENDIF
51.	
52.	Increment nilai i
53.	END FOR
54.	
55.	

Tabel 5.6 Algoritme mengajukan permohonan untuk Dokter

No	Pseudocode
56.	Simpan nilai variabel yang menyimpan nilai input user ke
57.	variabel array
58.	
59.	Panggil fungsi untuk menyimpan data ke model dan simpan
60.	nilai kembalian ke variabel
61.	
62.	IF nilai kembalian 'success'
63.	Panggil fungsi alert sukses
64.	ELSE panggil fungsi alert gagal
65.	ENDIF
66.	
67.	Arahkan ke halaman detil pengajuan
68.	
69.	Selesai

Berikut pada Tabel 5.5 adalah algoritme yang dijalankan jika Tenaga Kesehatan yang akan melakukan pengajuan teridentifikasi sebagai kelompok Perawat.

Tabel 5.7 Algoritme mengajukan permohonan untuk Perawat

No	Pseudocode
1.	Mulai
2.	Simpan asal lulusan ke variabel lokal
3.	Simpan tahun lulus ke variabel lokal
4.	Simpan nomor sib ke variabel lokal
5.	Simpan alamat praktik ke variabel lokal
6.	Simpan nama sarana ke variabel lokal
7.	Simpan alamat sarana ke variabel lokal
8.	Simpan nomor telepon sarana ke variabel lokal
9.	Panggil fungsi id_generator
10.	
11.	Atur konfigurasi tipe file upload
12.	Atur konfigurasi ukuran maksimal file upload
13.	
14.	For i=1; i<=6; 1++
15.	Switch i
16.	Jika i = 1, maka simpan letak penyimpanan berkas str
17.	ke variabel lokal
18.	Jika i = 2, maka simpan letak penyimpanan berkas SIP
19.	atau STR ke variabel lokal
20.	Jika i = 3, maka simpan letak penyimpanan berkas Surat
21.	Keterangan Sehat ke variabel lokal
22.	Jika i = 4, maka simpan letak penyimpanan berkas Surat
23.	Keterangan Pimpinan ke variabel lokal
24.	Jika i = 5, maka simpan letak penyimpanan berkas sipa
25.	1 ke variabel lokal
26.	End switch
27.	
28.	Simpan nama file upload ke variabel lokal
29.	Simpan nama baru file ke variabel lokal
30.	Atur konfigurasi nama file dengan nama baru
31.	Atur konfigurasi letak penyimpanan dengan nilai variabel
32.	lokal yang menyimpan letak penyimpanan file

Tabel 5.7 Algoritme mengajukan permohonan untuk Perawat

No	Pseudocode
33.	Inisialisasi konfigurasi
34.	
35.	IF file upload tidak kosong
36.	Simpan data file upload ke variabel array
37.	Simpan letak penyimpanan file ke variabel array
38.	ELSE isikan null ke variabel array penyimpan letak file
39.	ENDIF
40.	
41.	Increment nilai i
42.	END FOR
43.	
44.	Simpan nilai variabel yang menyimpan nilai input user ke variabel array
45.	
46.	
47.	Panggil fungsi untuk menyimpan data ke model dan simpan nilai kembalian ke variabel
48.	
49.	
50.	IF nilai kembalian 'success'
51.	Panggil fungsi alert sukses
52.	ELSE panggil fungsi alert gagal
53.	ENDIF
54.	
55.	Arahkan ke halaman detil pengajuan
56.	
57.	Selesai

Berikut pada Tabel 5.6 adalah algoritme yang dijalankan jika Tenaga Kesehatan yang akan melakukan pengajuan teridentifikasi sebagai kelompok Tenaga Teknis Kefarmasian.

Tabel 5.8 Algoritme mengajukan permohonan untuk Tenaga Teknis Kefarmasian

No	Pseudocode
1.	Mulai
2.	Simpan nomor strttk ke variabel lokal
3.	Simpan masa berlaku stra ke variabel lokal
4.	Simpan pendidikan akhir ke variabel lokal
5.	Simpan pengajuan surat ke variabel lokal
6.	Simpan nama fasilitas ke variabel lokal
7.	Simpan alamat fasilitas ke variabel lokal
8.	Simpan jadwal praktik ke variabel lokal
9.	Panggil fungsi id_generator
10.	
11.	Atur konfigurasi tipe file upload
12.	Atur konfigurasi ukuran maksimal file upload
13.	
14.	For i=1; i<=6; 1++
15.	Switch i
16.	Jika i = 1, maka simpan letak penyimpanan berkas strttk ke variabel lokal
17.	
18.	Jika i = 2, maka simpan letak penyimpanan berkas Pernyataan Apoteker ke variabel lokal
19.	

Tabel 5.8 Algoritme mengajukan permohonan untuk Tenaga Teknis Kefarmasian

No	Pseudocode
20.	Jika i = 3, maka simpan letak penyimpanan berkas surat
21.	persetujuan atasan ke variabel lokal
22.	Jika i = 4, maka simpan letak penyimpanan berkas surat
23.	rekomendasi organisasi ke variabel lokal
24.	Jika i = 5, maka simpan letak penyimpanan berkas sipttk
25.	1 ke variabel lokal
26.	Jika i = 6, maka simpan letak penyimpanan berkas sipttk
27.	2 ke variabel lokal
28.	End switch
29.	
30.	Simpan nama file upload ke variabel lokal
31.	Simpan nama baru file ke variabel lokal
32.	Atur konfigurasi nama file dengan nama baru
33.	Atur konfigurasi letak penyimpanan dengan nilai variabel
34.	lokal yang menyimpan letak penyimpanan file
35.	Inisialisasi konfigurasi
36.	
37.	IF file upload tidak kosong
38.	Simpan data file upload ke variabel array
39.	Simpan letak penyimpanan file ke variabel array
40.	ELSE isikan null ke variabel array penyimpan letak file
41.	ENDIF
42.	
43.	Increment nilai i
44.	END FOR
45.	
46.	Simpan nilai variabel yang menyimpan nilai input user ke
47.	variabel array
48.	
49.	Panggil fungsi untuk menyimpan data ke model dan simpan
50.	nilai kembalian ke variabel
51.	
52.	IF nilai kembalian 'success'
53.	Panggil fungsi alert sukses
54.	ELSE panggil fungsi alert gagal
55.	ENDIF
56.	
57.	Arahkan ke halaman detail pengajuan
58.	
59.	Selesai
60.	

5.6.2 Algoritme menyetujui pengajuan permohonan

Algoritme *function* kirim_persetujuan_pp() merupakan gambaran fungsi sistem untuk mengirimkan tanggapan dari Petugas Proses terhadap permohonan surat izin yang masuk ke *website* SINKES. Permohonan yang disetujui oleh Petugas Proses akan diteruskan hingga surat izin siap terbit, namun jika permohonan ditolak maka tidak akan ada penanganan lebih lanjut dari petugas Dinkes yang lain. Tabel 5.7 merupakan algoritme dari aktivitas menyetujui pengajuan permohonan.

Tabel 5.9 Algoritme menyetujui pengajuan permohonan

No	Pseudocode
1.	Mulai
2.	Simpan keputusan ke variabel lokal
3.	Simpan informasi tambahan ke variabel lokal
4.	Simpan informasi kelengkapan berkas ke variabel lokal
5.	Simpan informasi kekurangan ke variabel lokal
6.	
7.	Simpan nilai variabel yang menyimpan nilai input kedalam variabel array
8.	
9.	
10.	Panggil fungsi di model untuk menyimpan variabel array dan simpan kembaliannya dalam sebuah variabel
11.	
12.	
13.	IF nilai variabel kembalian 'success' atau true
14.	Panggil fungsi untuk mengembalikan alert sukses
15.	ELSE panggil fungsi untuk mengembalikan alert gagal
16.	
17.	Arahkan ke halaman detil pengajuan
18.	
19.	Selesai

5.6.3 Algoritme membuat jadwal kunjungan

Algoritme *function* `buat_jadwal` merupakan gambaran fungsi sistem untuk menangkap jadwal kunjungan yang dibuat oleh Kepala Bidang. Tabel 5.8 merupakan algoritme dari aktivitas membuat jadwal kunjungan.

Tabel 5.10 Algoritme membuat jadwal kunjungan pada kelas C_jadwal_kunjungan

No	Pseudocode
1.	Mulai
2.	Simpan tanggal kunjungan ke variabel lokal
3.	Simpan waktu kunjungan ke variabel lokal
4.	Simpan catatan tambahan ke variabel lokal
5.	
6.	Panggil fungsi di model untuk menyimpan tanggal dan waktu kunjungan array dan simpan kembaliannya dalam sebuah variabel
7.	
8.	
9.	
10.	Arahkan ke fungsi pembuatan jadwal kabid di C_penanganan
11.	
12.	Selesai

5.6.4 Algoritme menugaskan pembuatan surat tugas

Algoritme *function* `irim_persetujuan_kabid` merupakan gambaran fungsi sistem untuk mengolah masukan dari view yang digunakan untuk mengambil dan mengirimkan masukan terkait penugasan pencetakan surat izin dari Kepala Bidang ke Petugas Proses. Tabel 5.10 berikut menunjukkan algoritme menugaskan pembuatan surat izin.

Tabel 5.11 Algoritme menugaskan pembuatan surat izin

No	Pseudocode
1.	Mulai
2.	Simpan info tambahan kunjungan ke varabel lokal
3.	
4.	Simpan nilai variabel yang menyimpan nilai input kedalam
5.	variabel array
6.	
7.	Panggil fungsi di model untuk menyimpan tanggal dan waktu
8.	kunjungan array dan simpan kembaliannya dalam sebuah
9.	variabel
10.	
11.	IF nilai variabel kembalian 'success' atau true
12.	Panggil fungsi untuk mengembalikan alert sukses
13.	ELSE panggil fungsi untuk mengembalikan alert gagal
14.	
15.	
16.	Arahkan ke fungsi tampil halaman detil permohonan
17.	
18.	Selesai



BAB 6 IMPLEMENTASI

6.1 Implementasi program

Pada tahap ini, algoritme pemrograman yang telah dirancang pada poin 5.6 Bab Perancangan Sistem diubah kedalam bahasa pemrograman PHP untuk mewujudkannya menjadi fungsional yang bisa dijalankan.

6.1.1 Implementasi algoritme mengajukan permohonan

Implementasi algoritme dibuat sesuai dengan rancangan algoritme yang telah dibuat sebelumnya. *Function* ajukan_izin terdapat pada *controller* C_pengajuan yang memiliki fungsi untuk menerima masukan berupa id kelompok Tenaga Kesehatan yang sedang mengirimkan permohonan. Masukan yang didapat oleh *function* tersebut akan digunakan sebagai parameter untuk memanggil fungsi pengajuan izin sesuai dengan kelompok Tenaga Kesehatan yang bersangkutan. Untuk dapat mengajukan permohonan, maka Tenaga Kesehatan harus dalam keadaan *login* ke sistem.

Hasil algoritme untuk mengajukan permohonan dipisah menjadi 6 tabel yaitu Tabel 6.1, 6.2, 6.3, 6.4, 6.5 dan 6.6. Tabel 6.1 berisi *function* yang akan dipanggil pertama saat Tenaga Kesehatan menekan tombol untuk mengajukan permohonan, selanjutnya *function* tersebut akan memanggil *function* lain sesuai dengan kelompok Tenaga Kesehatan yang bersangkutan untuk pemrosesan selanjutnya. Sedangkan Tabel 6.2 sampai 6.6 merupakan *function* yang telah sesuai dengan kebutuhan pemrosesan data dari tiap kelompok Tenaga Kesehatan.

Tabel 6.1 Implementasi algoritme mengajukan permohonan

No	Kode Program
1.	function ajukan_izin(\$id_kelompok) {
2.	switch (\$id_kelompok) {
3.	case 1:
4.	\$this->ajukan_sipa_kefarmasian();;
5.	break;
6.	case 2:
7.	\$this->ajukan_sip_bidan();
8.	break;
9.	case 3:
10.	\$this->ajukan_sip_dokter();
11.	break;
12.	case 4:
13.	\$this->ajukan_sik_perawat();
14.	break;
15.	case 5:
16.	\$this->ajukan_sip_tt_kefarmasian();
17.	break;
18.	}
19.	}

Berikut pada Tabel 6.2 adalah algoritme yang dijalankan jika Tenaga Kesehatan yang akan melakukan pengajuan teridentifikasi sebagai kelompok Apoteker

Tabel 6.2 Implementasi algoritme mengajukan permohonan Apoteker

No	Kode Program
1.	/*=====*/
2.	/*fungsi pengajuan permohonan untuk Apoteker*/
3.	/*=====*/
4.	function ajukan_sipa_kefarmasian() {
5.	\$in_no_stra = \$this->input->post('in_no_stra');
6.	\$in_masa_stra = \$this->input->post('in_masa_stra');
7.	\$in_pendidikan_terakhir = \$this->input->
8.	post('in_pend_akhir');
9.	\$in_pengajuan_ke = \$this->input->post('in_sipa_ke');
10.	\$in_nama_fas = \$this->input->
11.	post('in_nama_fasilitas');
12.	\$in_alm_fas = \$this->input->post('in_alm_fasilitas');
13.	\$jadwal_praktik = \$this->input->
14.	post('send_data_jadwal');
15.	\$id_pengajuan = \$this->id_generator('SIPA-FAR');
16.	
17.	\$config['allowed_types'] = 'jpeg jpg png pdf';
18.	\$config['max_size'] = 3072;
19.	
20.	for (\$i=1; \$i<=6 ; \$i+1) {
21.	switch (\$i) {
22.	case '1':
23.	\$path = './file_upload/nakes_apoteker/scan
24.	STRA/';
25.	break;
26.	
27.	case '2':
28.	\$path = './file_upload/nakes_apoteker/scan
29.	memiliki tmp prak/';
30.	break;
31.	
32.	case '3':
33.	\$path = './file_upload/nakes_apoteker/scan
34.	pernyataan atasan/';
35.	break;
36.	
37.	case '4':
38.	\$path = './file_upload/nakes_apoteker/scan
39.	rekom/';
40.	break;
41.	
42.	case '5':
43.	if (isset(\$_FILES['in_sy'][\$i])) {
44.	\$path = './file_upload/nakes_apoteker/scan sipa
45.	1/';
46.	break;
47.	}
48.	
49.	case '6':
50.	if (isset(\$_FILES['in_sy'][\$i])) {
51.	\$path = './file_upload/nakes_apoteker/scan
52.	sipa 2/';
53.	break;
54.	}
55.	}
56.	}

Tabel 6.2 Implementasi algoritme mengajukan permohonan Apoteker

No	Kode Program
57.	\$nama_file = \$_FILES['in_sy'][\$i]['name'];
58.	\$nama_baru = \$id_pengajuan.'_'.\$nama_file;
59.	\$config['file_name'] = \$nama_baru;
60.	\$config['upload_path'] = \$path;
61.	\$this->upload->initialize(\$config);
62.	
63.	if (\$this->upload->do_upload('in_sy'.\$i) != "") {
64.	\$file_sy = array(
65.	'sy'.\$i => \$this->upload->data()
66.);
67.	\$upload_data['sy'.\$i] =
68.	"../".\$path.\$file_sy['sy'.\$i]['file_name'];
69.	} else {
70.	\$upload_data['sy'.\$i] = NULL;
71.	}
72.	\$i = \$i+1;
73.	}
74.	\$insert_data = array(
75.	'id_pengajuan' => \$id_pengajuan,
76.	'nik_nakes' => \$_SESSION['nik'],
77.	'id_status' => 1,
78.	'no_stra' => \$in_no_stra,
79.	'masa_stra' => \$in_masa_stra,
80.	'pend_akhir' => \$in_pendidikan_terakhir,
81.	'pengajuan_ke' => \$in_pengajuan_ke,
82.	'nama_fasilitas' => \$in_nama_fas,
83.	'alamat_fasilitas' => \$in_alm_fas,
84.	'jadwal_praktik' => \$jadwal_praktik,
85.	'data_upload' => \$upload_data
86.);
87.	\$execute = \$this->M_pengajuan->
88.	set_pengajuan_SIPA_farmasi(\$insert_data);
89.	
90.	if (\$execute == 'success') {
91.	\$this->Sys_notif->success("Permohonan Surat
92.	Izin berhasil diajukan.");
93.	} else {
94.	\$this->Sys_notif->danger("Permohonan Surat
95.	Izin gagal diajukan.");
96.	}
97.	redirect(base_url('halaman-pengajuan-izin/1'));
98.	}

Berikut pada Tabel 6.3 adalah algoritme yang dijalankan jika Tenaga Kesehatan yang akan melakukan pengajuan teridentifikasi sebagai kelompok Bidan

Tabel 6.3 Implementasi algoritme mengajukan permohonan Bidan

No	Kode Program
1.	/*=====*/
2.	/*fungsi pengajuan permohonan untuk Bidan*/
3.	/*=====*/
4.	function ajukan_sip_bidan(){
5.	
6.	\$in_thn_lulus = \$this->input->post('in_thn_lulus');
7.	\$in_sib = \$this->input->post('in_sib');

Tabel 6.3 Implementasi algoritme mengajukan permohonan Bidan

No	Kode Program
8.	\$in_sipb = \$this->input->post('in_sipb');
9.	\$in_alamat_praktik = \$this->input->
10.	>post('in_alamat_praktik');
11.	\$id_pengajuan = \$this->id_generator('SIP-BDN');
12.	\$config['allowed_types'] = 'jpeg jpg png pdf';
13.	\$config['max_size'] = 3072;
14.	
15.	for (\$i=1; \$i<=13 ; \$i+1) {
16.	switch (\$i) {
17.	case '1':
18.	\$path = './file_upload/nakes_bidan/KTP/';
19.	break;
20.	case '2':
21.	\$path =
22.	'./file_upload/nakes_bidan/Rekomendasi_Puskesmas_Wilayah/
23.	';
24.	break;
25.	case '3':
26.	\$path =
27.	'./file_upload/nakes_bidan/Surat_Kesanggupan_Dokter_Penga
28.	was/';
29.	break;
30.	case '4':
31.	\$path =
32.	'./file_upload/nakes_bidan/Surat_Rekomendasi_Aatasan/';
33.	break;
34.	case '5':
35.	\$path =
36.	'./file_upload/nakes_bidan/Rekomendasi_IBI/';
37.	break;
38.	case '6':
39.	\$path = './file_upload/nakes_bidan/STR
40.	berlaku_dan_terlegalisir/';
41.	break;
42.	case '7':
43.	\$path =
44.	'./file_upload/nakes_bidan/SK_Pengangkatan-Penempatan/';
45.	break;
46.	case '8':
47.	\$path =
48.	'./file_upload/nakes_bidan/Daftar_Peralatan/';
49.	break;
50.	case '9':\$path =
51.	'./file_upload/nakes_bidan/SIPB_Lama/';
52.	break;
53.	case '10':
54.	\$path =
55.	'./file_upload/nakes_bidan/Gambar_Denah_Bangunan/';
56.	break;
57.	case '11':
58.	\$path = './file_upload/nakes_bidan/Surat_Sihat/';
59.	break;
60.	case '12':
61.	\$path =
62.	'./file_upload/nakes_bidan/Surat_Pernyataan_Kepemilikan/
63.	';

Tabel 6.3 Implementasi algoritme mengajukan permohonan Bidan

No	Kode Program
64.	break;
65.	case '13':
66.	\$path =
67.	'./file_upload/nakes_bidan/Surat_Bidan_Mitra/';
68.	break;
69.	}
70.	
71.	\$nama_file = \$_FILES['in_sy'][\$i]['name'];
72.	\$nama_baru = \$id_pengajuan.'_'. \$nama_file;
73.	\$config['file_name'] = \$nama_baru;
74.	\$config['upload_path'] = \$path;
75.	\$this->upload->initialize(\$config);
76.	
77.	if (\$this->upload->do_upload('in_sy'.\$i) != "") {
78.	\$file_sy = array(
79.	'sy'.\$i => \$this->upload->data()
80.);
81.	\$upload_data['sy'.\$i]
82.	"../".\$path.\$file_sy['sy'.\$i]['file_name'];
83.	} else {
84.	\$upload_data['sy'.\$i] = NULL;
85.	}
86.	
87.	\$i = \$i+1;
88.	}
89.	\$insert_data = array(
90.	'id_pengajuan' => \$id_pengajuan,
91.	'nik_nakes' => \$_SESSION['nik'],
92.	'id_status' => 1,
93.	'thn_lulus' => \$in_thn_lulus,
94.	'no_sib' => \$in_sib,
95.	'no_sipb' => \$in_sipb,
96.	'alamat_praktik' => \$in_alamat_praktik,
97.	'data_upload' => \$upload_data
98.);
99.	\$execute = \$this->M_pengajuan-
100.	>set_pengajuan_SIP_bidan(\$insert_data);
101.	
102.	if (\$execute == 'success') {
103.	\$this->Sys_notif->success("Permohonan Surat Izin
104.	berhasil diajukan.");
105.	} else {
106.	\$this->Sys_notif->danger("Permohonan Surat Izin
107.	gagal diajukan.");
108.	}
109.	redirect(base_url('halaman-pengajuan-izin/2'));
110.	}

Berikut pada Tabel 6.4 adalah algoritme yang dijalankan jika Tenaga Kesehatan yang akan melakukan pengajuan teridentifikasi sebagai kelompok Bidan

Tabel 6.4 Implementasi algoritme mengajukan permohonan Dokter

No	Kode Program
1.	/*=====*/
2.	/*fungsi pengajuan permohonan untuk Dokter*/

Tabel 6.4 Implementasi algoritme mengajukan permohonan Dokter

No	Kode Program
3.	/*=====*/
4.	function ajukan_sip_dokter() {
5.	\$in_lulusan = \$this->input->post('in_lulusan');
6.	\$in_thn_lulus = \$this->input->post('in_thn_lulus');
7.	\$in_str = \$this->input->post('in_str');
8.	\$in_sip = \$this->input->post('in_sip');
9.	\$in_unit_kerja = \$this->input->post('in_unit_kerja');
10.	\$in_idi = \$this->input->post('in_idi');
11.	\$no_rekom = \$this->input->post('in_no_rekom');
12.	\$id_pengajuan = \$this->id_generator('SIP-DOK');
13.	
14.	\$config['allowed_types'] = 'jpeg jpg png pdf';
15.	\$config['max_size'] = 3072;
16.	
17.	for (\$i=1; \$i<=10 ; \$i+1) {
18.	switch (\$i) {
19.	case '1':
20.	\$path = './file_upload/nakes_dokter/KTP/';
21.	break;
22.	case '2':
23.	\$path = './file_upload/nakes_dokter/STR/';
24.	break;
25.	case '3':
26.	\$path
27.	'./file_upload/nakes_dokter/SK_Pengangkatan_Awal-
28.	Akhir/';
29.	break;
30.	case '4':
31.	\$path = './file_upload/nakes_dokter/SK_Pegawai/';
32.	break;
33.	case '5':
34.	\$path = './file_upload/nakes_dokter/SK_Pensiun/';
35.	break;
36.	case '6':
37.	\$path
38.	'./file_upload/nakes_dokter/Surat_Persetujuan_Aatasan/';
39.	break;
40.	case '7':
41.	\$path
42.	'./file_upload/nakes_dokter/Surat_Memiliki_Tempat_Prakti
43.	k/';
44.	break;
45.	case '8':
46.	\$path = './file_upload/nakes_dokter/Surat_Tempat
47.	Sarana Pratik/';
48.	break;
49.	case '9':
50.	\$path
51.	'./file_upload/nakes_dokter/Surat_Rekomendasi/';
52.	break;
53.	case '10':
54.	\$path
55.	'./file_upload/nakes_dokter/Surat_Kadin_Asal/';
56.	break;
57.	}
58.	\$nama_file = \$_FILES['in_sy'][\$i]['name'];

Tabel 6.4 Implementasi algoritme mengajukan permohonan Dokter

No	Kode Program
59.	\$nama_baru = \$id_pengajuan.'_'.\$nama_file;
60.	\$config['file_name'] = \$nama_baru;
61.	\$config['upload_path'] = \$path;
62.	\$this->upload->initialize(\$config);
63.	
64.	if (\$this->upload->do_upload('in_sy'.\$i) != "") {
65.	\$file_sy = array(
66.	'sy'.\$i => \$this->upload->data()
67.);
68.	\$upload_data['sy'.\$i] =
69.	"../".\$path.\$file_sy['sy'.\$i]['file_name'];
70.	} else {
71.	\$upload_data['sy'.\$i] = NULL;
72.	}
73.	\$i = \$i+1;
74.	}
75.	
76.	\$insert_data = array(
77.	'id_pengajuan' => \$id_pengajuan,
78.	'nik_nakes' => \$_SESSION['nik'],
79.	'id_status' => 1,
80.	'lulusan' => \$in_lulusan,
81.	'thn_lulus' => \$in_thn_lulus,
82.	'no_str' => \$in_str,
83.	'no_sip' => \$in_sip,
84.	'unit_kerja' => \$in_unit_kerja,
85.	'idi' => \$in_idi,
86.	'no_rekom' => \$no_rekom,
87.	'data_upload' => \$upload_data
88.);
89.	
90.	\$execute = \$this->M_pengajuan->
91.	set_pengajuan_SIP_dokter(\$insert_data);
92.	
93.	if (\$execute == 'success') {
94.	\$this->Sys_notif->success("Permohonan Surat Izin
95.	berhasil diajukan.");
96.	} else {
97.	\$this->Sys_notif->danger("Permohonan Surat Izin
98.	gagal diajukan.");
99.	}
100.	redirect(base_url('halaman-pengajuan-izin/3'));
101.	}

Berikut pada Tabel 6.5 adalah algoritme yang dijalankan jika Tenaga Kesehatan yang akan melakukan pengajuan teridentifikasi sebagai kelompok Bidan

Tabel 6.5 Implementasi algoritme mengajukan permohonan untuk Perawat

No	Kode Program
1.	/*=====*/
2.	/*fungsi pengajuan permohonan untuk Perawat*/
3.	/*=====*/
4.	function ajukan_sik_perawat(){
5.	\$in_lulusan = \$this->input->post('in_lulusan');
6.	\$in_thn_lulus = \$this->input->post('in_thn_lulus');

Tabel 6.5 Implementasi algoritme mengajukan permohonan untuk Perawat

No	Kode Program
7.	<code>\$in_sib = \$this->input->post('in_sib');</code>
8.	<code>\$in_alamat_praktik = \$this->input-</code>
9.	<code>>post('in_alamat_praktik');</code>
10.	<code>\$in_nama_sarana = \$this->input-</code>
11.	<code>>post('in_nama_sarana');</code>
12.	<code>\$in_alm_sarana = \$this->input->post('in_alm_sarana');</code>
13.	<code>\$in_tel_sarana = \$this->input->post('in_tel_sarana');</code>
14.	<code>\$id_pengajuan = \$this->id_generator('SIP-PRW');</code>
15.	
16.	<code>\$config['allowed_types'] = 'jpeg jpg png pdf';</code>
17.	<code>\$config['max_size'] = 3072;</code>
18.	
19.	<code>for (\$i=1; \$i<=5 ; \$i+1) {</code>
20.	<code> switch (\$i) {</code>
21.	<code> case '1':</code>
22.	<code> \$path = './file_upload/nakes_perawat/KTP/';</code>
23.	<code> break;</code>
24.	<code> case '2':</code>
25.	<code> \$path =</code>
26.	<code> './file_upload/nakes_perawat/Surat_SIP_STR/';</code>
27.	<code> break;</code>
28.	<code> case '3':</code>
29.	<code> \$path =</code>
30.	<code> './file_upload/nakes_perawat/Surat_Keterangan_Sihat/';</code>
31.	<code> break;</code>
32.	<code> case '4':</code>
33.	<code> \$path =</code>
34.	<code> './file_upload/nakes_perawat/Surat_Keterangan_Pimpinan/';</code>
35.	<code> break;</code>
36.	<code> case '5':</code>
37.	<code> \$path =</code>
38.	<code> './file_upload/nakes_perawat/Surat_Rekomendasi_Organisasi</code>
39.	<code> _Profesi/';</code>
40.	<code> break;</code>
41.	<code> }</code>
42.	
43.	<code>\$nama_file = \$_FILES['in_sy'][\$i]['name'];</code>
44.	<code>\$nama_baru = \$id_pengajuan.'_'.\$nama_file;</code>
45.	
46.	<code>\$config['file_name'] = \$nama_baru;</code>
47.	<code>\$config['upload_path'] = \$path;</code>
48.	<code>\$this->upload->initialize(\$config);</code>
49.	
50.	<code>if (\$this->upload->do_upload('in_sy'.\$i) != "") {</code>
51.	<code> \$file_sy = array(</code>
52.	<code> 'sy'.\$i => \$this->upload->data()</code>
53.	<code>);</code>
54.	<code> \$upload_data['sy'.\$i] =</code>
55.	<code> "../".\$path.\$file_sy['sy'.\$i]['file_name'];</code>
56.	<code> } else {</code>
57.	<code> \$upload_data['sy'.\$i] = NULL;</code>
58.	<code> }</code>
59.	<code> \$i = \$i+1;</code>
60.	<code>}</code>
61.	
62.	<code>\$insert_data = array(</code>

Tabel 6.5 Implementasi algoritme mengajukan permohonan untuk Perawat

No	Kode Program
63.	'kode' => \$id_pengajuan,
64.	'id_status' => 1,
65.	'lulusan' => \$in_lulusan,
66.	'thn_lulus' => \$in_thn_lulus,
67.	'no_sip_str' => \$in_sib,
68.	'tmp_kerja' => \$in_alamat_praktik,
69.	'nama_sarana'=> \$in_nama_sarana,
70.	'alamat_sarana' => \$in_alm_sarana,
71.	'tlp_sarana' => \$in_tel_sarana,
72.	'data_upload' => \$upload_data
73.);
74.	
75.	\$execute = \$this->M_pengajuan-
76.	>set_pengajuan_SIK_perawat(\$insert_data);
77.	
78.	if (\$execute == 'success') {
79.	\$this->Sys_notif->success("Permohonan Surat Izin
80.	berhasil diajukan.");
81.	} else {
82.	\$this->Sys_notif->danger("Permohonan Surat Izin
83.	gagal diajukan.");
84.	}
85.	redirect(base_url('halaman-pengajuan-izin/4'));
86.	}

Berikut pada Tabel 6.6 adalah algoritme yang dijalankan jika Tenaga Kesehatan yang akan melakukan pengajuan teridentifikasi sebagai kelompok Bidan

Tabel 6.6 Implementasi algoritme mengajukan permohonan untuk Tenaga Teknis Kefarmasian

No	Kode Program
1.	/*=====*/
2.	/*fungsi pengajuan permohonan untuk Tenaga Teknis
3.	Kefarmasian*/
4.	/*=====*/
5.	function ajukan_sip_tt_kefarmasian(){
6.	\$in_strttk = \$this->input->post('in_no_strttk');
7.	\$in_masa_strttk = \$this->input-
8.	>post('in_masa_berlaku');
9.	\$in_pend_akhir = \$this->input-
10.	>post('in_pendidikan_terakhir');
11.	\$in_permohonan_ke = \$this->input-
12.	>post('in_permohonan_ke');
13.	\$in_nama_fasilitas = \$this->input-
14.	>post('in_nama_fasilitas_kefarmasian');
15.	\$in_alamat_fasilitas = \$this->input-
16.	>post('in_alamat');
17.	\$jadwal_praktik = \$this->input-
18.	>post('send_data_jadwal');
19.	\$id_pengajuan = \$this->id_generator('SIP-TTK');
20.	
21.	\$config['allowed_types'] = 'jpeg jpg png pdf';
22.	\$config['max_size'] = 3072;
23.	for (\$i=1; \$i<=6 ; \$i+1) {
24.	switch (\$i) {

Tabel 6.6 Implementasi algoritme mengajukan permohonan untuk Tenaga Teknis Kefarmasian

No	Kode Program
25.	case '1':
26.	\$path =
27.	'./file_upload/nakes_tt_kefarmasian/STRTTK/';
28.	break;
29.	case '2':
30.	\$path =
31.	'./file_upload/nakes_tt_kefarmasian/Pernyataan_Apoteker/';
32.	;
33.	break;
34.	case '3':
35.	\$path =
36.	'./file_upload/nakes_tt_kefarmasian/Surat_Persetujuan_Atasan/';
37.	break;
38.	case '4':
39.	\$path =
40.	'./file_upload/nakes_tt_kefarmasian/Surat_Rekomendasi_Organisasi/';
41.	break;
42.	case '5':
43.	\$path =
44.	'./file_upload/nakes_tt_kefarmasian/SIPTTK1/';
45.	break;
46.	case '6':
47.	\$path =
48.	'./file_upload/nakes_tt_kefarmasian/SIPTTK2/';
49.	break;
50.	}
51.	
52.	
53.	
54.	\$nama_file = \$_FILES['in_sy'][\$i]['name'];
55.	\$nama_baru = \$_SESSION['nik'].'-'.\$nama_file;
56.	\$config['file_name'] = \$nama_baru;
57.	\$config['upload_path'] = \$path;
58.	\$this->upload->initialize(\$config);
59.	
60.	if (\$this->upload->do_upload('in_sy'.\$i) != "") {
61.	\$file_sy = array(
62.	'sy'.\$i => \$this->upload->data()
63.);
64.	\$upload_data['sy'.\$i] =
65.	"../".\$path.\$file_sy['sy'.\$i]['file_name'];
66.	} else {
67.	\$upload_data['sy'.\$i] = NULL;
68.	}
69.	\$i = \$i+1;
70.	}
71.	\$insert_data = array(
72.	'kode' => \$id_pengajuan,
73.	'id_status' => 1,
74.	'in_strttk' => \$in_strttk,
75.	'in_masa_strttk' => \$in_masa_strttk,
76.	'in_pend_akhir' => \$in_pend_akhir,
77.	'in_permohonan_ke' => \$in_permohonan_ke,
78.	'in_nama_fasilitas' =>
79.	\$in_nama_fasilitas,

Tabel 6.6 Implementasi algoritme mengajukan permohonan untuk Tenaga Teknis Kefarmasian

No	Kode Program
80.	'in_alamat_fasilitas' => \$in_alamat_fasilitas,
81.	'jadwal_praktik' => \$jadwal_praktik,
82.	'data_upload' => \$upload_data
83.);
84.	
85.	\$execute = \$this->M_pengajuan-
86.	>set_pengajuan_SIP_ttkefarmasian(\$insert_data);
87.	
88.	if (\$execute == 'success') {
89.	\$this->Sys_notif->success("Permohonan Surat Izin
90.	berhasil diajukan.");
91.	} else {
92.	\$this->Sys_notif->danger("Permohonan Surat Izin
93.	gagal diajukan.");
94.	}
95.	redirect(base_url('halaman-pengajuan-izin/5'));
96.	}

Pada Tabel 6.7 hingga 6.12 merupakan penjelasan kode program yang ada pada Tabel 6.1 hingga 6.6. Tabel 6.7 menjelaskan kode program Tabel 6.1 yang berisi *function* ajukan_izin. Tabel 6.8 menjelaskan kode program Tabel 6.2 yang berisi *function* mengajukan permohonan untuk Apoteker. Tabel 6.9 menjelaskan kode program Tabel 6.3 yang berisi *function* mengajukan permohonan untuk Bidan. Tabel 6.10 menjelaskan kode program Tabel 6.4 yang berisi *function* mengajukan permohonan untuk Dokter. Tabel 6.11 menjelaskan kode program Tabel 6.5 yang berisi *function* mengajukan permohonan untuk Perawat. Tabel 6.2 menjelaskan kode program Tabel 6.6 yang berisi *function* mengajukan permohonan untuk Tenaga Teknis Kefarmasian.

Tabel 6.7 Penjelasan implementasi algoritme mengajukan permohonan

Baris	Penjelasan
1	Melakukan deklarasi <i>function</i> dengan nama ajukan_izin dan dengan parameter id_kelompok
2	Seleksi kondisi <i>switch</i> dengan kondisi yang dicek adalah id_kelompok
3-5	Jalankan <i>function</i> ajukan_sipa_kefarmasian jika id_kelompok bernilai 1
6-8	Jalankan <i>function</i> ajukan_sip_bidan jika id_kelompok bernilai 2
9-11	Jalankan <i>function</i> ajukan_sip_dokter jika id_kelompok bernilai 3
12-14	Jalankan <i>function</i> ajukan_sik_perawat jika id_kelompok bernilai 4
15-17	Jalankan <i>function</i> ajukan_sip_tt_kefarmasian jika id_kelompok bernilai 5

Tabel 6.8 Penjelasan implementasi algoritme mengajukan permohonan Apoteker

Baris	Penjelasan
4	Melakukan deklarasi <i>function</i> dengan nama ajukan_sipa_kefarmasian
5-14	Mengambil nilai masukan dari view dan disimpan kedalam variabel
15	Memanggil <i>function</i> id_generator dengan parameter 'SIPA-Far'
17-18	Konfigurasi <i>library upload</i>

Tabel 6.8 Penjelasan implementasi algoritme mengajukan permohonan Apoteker

Baris	Penjelasan
20	Deklarasi perulangan <i>for</i> mulai 1 sampai kurang dari samadengan 6
21	Deklarasi seleksi kondisi <i>switch</i> dengan kondisi variabel <i>i</i> dari perulangan <i>for</i>
22-53	Kondisi yang akan dijalankan jika memenuhi kondisi variabel <i>i</i>
57-61	Konfigurasi <i>library upload</i>
63-70	Seleksi kondisi untuk memasukkan <i>path</i> berkas ke dalam <i>array</i>
72	Increment nilai <i>i</i>
74-85	Penyimpanan nilai yang diambil dari view kedalam <i>array</i> dengan nama <i>insert_data</i>
87-88	Memanggil fungsi <i>set_pengajuan_SIPA_farmasi</i> dengan memasukkan parameter <i>array insert_data</i>
91-95	Seleksi kondisi untuk memeriksa apakah proses memasukkan berhasil atau tidak untuk menentukan pesan kembalian ke <i>view</i>
97	Mengarahkan ke halaman pengajuan izin untuk Apoteker

Tabel 6.9 Penjelasan implementasi algoritme mengajukan permohonan Bidan

Baris	Penjelasan
4	Melakukan deklarasi <i>function</i> dengan nama <i>ajukan_sip_bidan</i>
6-10	Mengambil nilai masukan dari <i>view</i> dan disimpan kedalam variabel
11	Memanggil <i>function id_generator</i> dengan parameter 'SIP-BDN'
12-13	Konfigurasi <i>library upload</i>
15	Deklarasi perulangan <i>for</i> mulai 1 sampai kurang dari samadengan 13
16	Deklarasi seleksi kondisi <i>switch</i> dengan kondisi variabel <i>i</i> dari perulangan <i>for</i>
17-68	Kondisi yang akan dijalankan jika memenuhi kondisi variabel <i>i</i>
71-75	Konfigurasi <i>library upload</i>
77-84	Seleksi kondisi untuk memasukkan <i>path</i> berkas ke dalam <i>array</i>
87	Increment nilai <i>i</i>
89-97	Penyimpanan nilai yang diambil dari view kedalam <i>array</i> dengan nama <i>insert_data</i>
99-100	Memanggil fungsi <i>set_pengajuan_SIP_bidan</i> dengan memasukkan parameter <i>array insert_data</i>
102-107	Seleksi kondisi untuk memeriksa apakah proses memasukkan berhasil atau tidak untuk menentukan pesan kembalian ke <i>view</i>
109	Mengarahkan ke halaman pengajuan izin untuk Bidan

Tabel 6.10 Penjelasan implementasi algoritme mengajukan permohonan Dokter

Baris	Penjelasan
4	Melakukan deklarasi <i>function</i> dengan nama <i>ajukan_sip_dokter</i>
5-11	Mengambil nilai masukan dari <i>view</i> dan disimpan kedalam variabel
12	Memanggil <i>function id_generator</i> dengan parameter 'SIP-DOK'
14-15	Konfigurasi <i>library upload</i>
17	Deklarasi perulangan <i>for</i> mulai 1 sampai kurang dari samadengan 10
18	Deklarasi seleksi kondisi <i>switch</i> dengan kondisi variabel <i>i</i> dari perulangan <i>for</i>
19-56	Kondisi yang akan dijalankan jika memenuhi kondisi variabel <i>i</i>

Tabel 6.10 Penjelasan implementasi algoritme mengajukan permohonan Dokter

Baris	Penjelasan
58-62	Konfigurasi <i>library upload</i>
64-71	Seleksi kondisi untuk memasukkan <i>path</i> berkas ke dalam <i>array</i>
73	Increment nilai <i>i</i>
76-87	Penyimpanan nilai yang diambil dari <i>view</i> kedalam <i>array</i> dengan nama <i>insert_data</i>
90-91	Memanggil fungsi <i>set_pengajuan_SIP_dokter</i> dengan memasukkan parameter <i>array insert_data</i>
93-98	Seleksi kondisi untuk memeriksa apakah proses memasukkan berhasil atau tidak untuk menentukan pesan kembalin ke <i>view</i>
100	Mengarahkan ke halaman pengajuan izin untuk Dokter

Tabel 6.11 Penjelasan implementasi algoritme mengajukan permohonan Perawat

Baris	Penjelasan
4	Melakukan deklarasi <i>function</i> dengan nama <i>ajukan_sik_perawat</i>
5-13	Mengambil nilai masukan dari <i>view</i> dan disimpan kedalam variabel
14	Memanggil <i>function</i> <i>id_generator</i> dengan parameter 'SIK-PRW'
16-17	Konfigurasi <i>library upload</i>
19	Deklarasi perulangan <i>for</i> mulai 1 sampai kurang dari samadengan 5
20	Deklarasi seleksi kondisi <i>switch</i> dengan kondisi variabel <i>i</i> dari perulangan <i>for</i>
21-40	Kondisi yang akan dijalankan jika memenuhi kondisi variabel <i>i</i>
43-48	Konfigurasi <i>library upload</i>
50-57	Seleksi kondisi untuk memasukkan <i>path</i> berkas ke dalam <i>array</i>
59	Increment nilai <i>i</i>
62-72	Penyimpanan nilai yang diambil dari <i>view</i> kedalam <i>array</i> dengan nama <i>insert_data</i>
75-76	Memanggil fungsi <i>set_pengajuan_SIK_perawat</i> dengan memasukkan parameter <i>array insert_data</i>
78-83	Seleksi kondisi untuk memeriksa apakah proses memasukkan berhasil atau tidak untuk menentukan pesan kembalian ke <i>view</i>
85	Mengarahkan ke halaman pengajuan izin untuk Perawat

Tabel 6.12 Penjelasan implementasi algoritme mengajukan permohonan Tenaga Teknis Kefarmasian

Baris	Penjelasan
5	Melakukan deklarasi <i>function</i> dengan nama <i>ajukan_sip_tt_kefarmasian</i>
6-18	Mengambil nilai masukan dari <i>view</i> dan disimpan kedalam variabel
19	Memanggil <i>function</i> <i>id_generator</i> dengan parameter 'SIP-TTK'
21-22	Konfigurasi <i>library upload</i>
23	Deklarasi perulangan <i>for</i> mulai 1 sampai kurang dari samadengan 6
24	Deklarasi seleksi kondisi <i>switch</i> dengan kondisi variabel <i>i</i> dari perulangan <i>for</i>
25-51	Kondisi yang akan dijalankan jika memenuhi kondisi variabel <i>i</i>
54-58	Konfigurasi <i>library upload</i>
60-67	Seleksi kondisi untuk memasukkan <i>path</i> berkas ke dalam <i>array</i>
69	Increment nilai <i>i</i>

71-72	Penyimpanan nilai yang diambil dari view kedalam <i>array</i> dengan nama <i>insert_data</i>
85-86	Memanggil fungsi <i>set_pengajuan_SIP_tt_kefarmasian</i> dengan memasukkan parameter <i>array insert_data</i>
88-93	Seleksi kondisi untuk memeriksa apakah proses memasukkan berhasil atau tidak untuk menentukan pesan kembalian ke <i>view</i>
95	Mengarahkan ke halaman pengajuan izin untuk Tenaga Teknis Kefarmasian

6.1.2 Implementasi algoritme menyetujui pengajuan permohonan

Implementasi algoritme dibuat sesuai dengan rancangan algoritme yang telah dibuat sebelumnya. *Function* *kirim_persetujuan_pp* terdapat pada *controller* *C_penanganan* yang berfungsi untuk mengolah masukan dari formulir tindakan Petugas Proses ketika menentukan apakah sebuah permohonan akan disetujui untuk ditangani atau ditolak. Parameter pada *function* tersebut adalah *id_pengajuan* dari permohonan yang akan dilakukan penindakan. Untuk dapat menjalankan *function* ini, maka aktor harus *login* sebagai Petugas Proses.

Tabel 6.13 Implementasi algoritme menyetujui pengajuan permohonan

No	Kode Program
1.	<code>function kirim_persetujuan_pp(\$id_pengajuan){</code>
2.	<code> \$diterima = \$this->input-</code>
3.	<code> >post('in_diterima');</code>
4.	<code> \$info_tambahan = \$this->input->post('info-</code>
5.	<code> tambahan');</code>
6.	<code> \$grup_lengkap = \$this->input->post('in_lengkap');</code>
7.	<code> \$kekurangan_berkas = \$this->input-</code>
8.	<code> >post('kekurangan_berkas');</code>
9.	
10.	<code> \$data = array();</code>
11.	<code> \$raw_jenis = explode('-', \$id_pengajuan);</code>
12.	<code> \$raw_jenis_perizinan = \$raw_jenis[1];</code>
13.	<code> \$jenis_perizinan = null;</code>
14.	
15.	<code> switch (\$raw_jenis_perizinan) {</code>
16.	<code> case 'BDN':</code>
17.	<code> \$raw_data = array(</code>
18.	<code> array('Scan Kartu Tanda Penduduk (KTP) yang</code>
19.	<code> masih berlaku' => \$this->input->post('checklist_ktp')),</code>
20.	<code> array('Scan Surat Rekomendasi Puskesmas di</code>
21.	<code> wilayah tempat melakukan Praktik Bidan' => \$this->input-</code>
22.	<code> >post('checklist_rekom_pkm_wil_prak_bidan')),</code>
23.	<code> array('Scan Surat Kesanggupan sebagai Dokter</code>
24.	<code> Pengawas dari seorang Dokter (secara perorangan) yang telah</code>
25.	<code> memiliki Surat Ijin Praktik (SIP)' => \$this->input-</code>
26.	<code> >post('checklist_sk_kesanggupan_dokter')),</code>
27.	<code> array('Scan Surat Rekomendasi dari Atasan</code>
28.	<code> Langsung, bila dalam pelaksanaan Masa Bhakti atau sebagai</code>
29.	<code> Pegawai Negeri atau Pegawai pada Sarana Kesehatan Swasta'</code>
30.	<code> => \$this->input->post('checklist_rekom_atasan')),</code>
31.	<code> array('Scan Rekomendasi dari Ikatan Bidan</code>
32.	<code> Indonesia (IBI)' => \$this->input-</code>
33.	<code> >post('checklist_rekom_ibi')),</code>
34.	<code> }</code>

Tabel 6.13 Implementasi algoritme menyetujui pengajuan permohonan

No	Kode Program
35.	array('Scan STR yang masih berlaku' => \$this->input->post('checklist_str')),
36.	
37.	array('Scan Surat Keputusan
38.	Pengangkatan/Penempatan pertama dan terakhir yang bekerja
39.	pada Unit Pelayanan baik Pemerintah/Swasta' => \$this->input->post('checklist_sk_pengangkatan_kerja')),
40.	
41.	array('Daftar peralatan yang dimiliki untuk
42.	Praktik Bidan' => \$this->input->post('checklist_daftar_peralatan_bidan')),
43.	
44.	array('Scan SIPB yang lama, apabila mengajukan
45.	permohonan SIPB untuk pindah alamat dan perpanjangan' =>
46.	\$this->input->post('checklist_scan_sipb')),
47.	array('Gambar Denah Bangunan/Ruangan yang
48.	menggambarkan fungsi serta ukuran luas dan peta lokasi' =>
49.	\$this->input->post('checklist_gambar_denah_bangunan')),
50.	array('Scan Surat Keterangan Sehat dari Dokter
51.	yang telah memiliki Surat Ijin Praktik (SIP)' => \$this->input->post('checklist_sk_sehat')),
52.	
53.	array('Scan Surat pernyataan memiliki tempat
54.	praktik' => \$this->input->post('checklist_pernyataan_tempat_praktik')),
55.	
56.	array('Scan Surat pernyataan bidan mitra' =>
57.	\$this->input->post('checklist_pernyataan_bidan_mitra')),
58.);
59.	array_push(\$data, \$raw_data);
60.	break;
61.	
62.	
63.	case 'DOK':
64.	\$raw_data = array(
65.	array('Scan Kartu Tanda Penduduk (KTP) yang
66.	masih berlaku' => \$this->input->post('checklist_ktp')),
67.	array('Scan STR yang diterbitkan oleh KKI yang
68.	masih berlaku.' => \$this->input->post('checklist_str')),
69.	array('Scan SK Pertama dan SK Pangkat Terakhir
70.	bagi PNS.' => \$this->input->post('checklist_sk_pangkat_terakhir')),
71.	
72.	array('Scan SK pengangkatan sebagai pegawai
73.	honorer atau pegawai Tetap yang bekerja di sarana swasta.'
74.	=> \$this->input->post('checklist_sk_pegawai')),
75.	array('Scan SK Pensiun bagi yang purna tugas.'
76.	=> \$this->input->post('checklist_sk_pensiun')),
77.	array('Surat persetujuan dari atasan langsung
78.	bagi Dokter dan Dokter Gigi yang bekerja pada
79.	instansi/fasilitas pelayanan kesehatan pemerintah atau pada
80.	instansi/fasilitas pelayanan kesehatan lain secara purna
81.	waktu.' => \$this->input->post('checklist_persetujuan_atasan')),
82.	
83.	array('Scan Surat pernyataan mempunyai tempat
84.	praktik, atau' => \$this->input->post('checklist_pernyataan_tempat_praktik')),
85.	
86.	array('Scan surat keterangan dari fasilitas
87.	pelayanan kesehatan sebagai tempat praktik.' => \$this->input->post('checklist_keterangan_faskes')),
88.	
89.	
90.	

Tabel 6.13 Implementasi algoritme menyetujui pengajuan permohonan

No	Kode Program
91.	array('Scan Surat Rekomendasi dari Organisasi
92.	Profesi di wilayah tempat praktik yang masih berlaku.' =>
93.	\$this->input->post('checklist_rekom_org_profesi')),
94.	array('Scan Surat keterangan dari Kepala Dinas
95.	Kesehatan tempat asal (jika tempat kerja pemohon berada di
96.	luar Kota Malang atau mempunyai tempat praktik di luar Kota
97.	Malang).' => \$this->input-
98.	>post('checklist_ket_kepala_dinas_luar')),
99.);
100.	array_push(\$data, \$raw_data);
101.	break;
102.	
103.	
104.	case 'PRW':
105.	\$raw_data = array(
106.	array('Scan Kartu Tanda Penduduk (KTP) yang
107.	masih berlaku' => \$this->input->post('checklist_ktp')),
108.	array('Scan SIP/STR yang masih berlaku.' =>
109.	\$this->input->post('checklist_sip_str')),
110.	array('Scan surat keterangan sehat dari dokter
111.	yang memiliki SIP' => \$this->input-
112.	>post('checklist_sk_sehat')),
113.	array('Scan surat Keterangan dari pimpinan
114.	sarana pelayanan kesehatan' => \$this->input-
115.	>post('checklist_sk_pimpinan_faskes')),
116.	array('Scan Surat Rekomendasi dari Organisasi
117.	Profesi' => \$this->input-
118.	>post('checklist_rekom_org_profesi')),
119.);
120.	array_push(\$data, \$raw_data);
121.	break;
122.	
123.	case 'TTK':
124.	\$raw_data = array(
125.	array('Scan STRTTK.' => \$this->input-
126.	>post('checklist_scan_strttk')),
127.	array('Scan Surat pernyataan Apoteker atau
128.	pimpinan tempat Pemohon melaksanakan Pekerjaan
129.	Kefarmasian.' => \$this->input-
130.	>post('checklist_pernyataan_pimpinan')),
131.	array('Scan Surat Persetujuan Atasan Langsung.'
132.	=> \$this->input->post('checklist_persetujuan_atasan')),
133.	array('Scan Surat rekomendasi dari organisasi
134.	profesi.' => \$this->input-
135.	>post('checklist_rekom_org_profesi')),
136.);
137.	array_push(\$data, \$raw_data);
138.	break;
139.	
140.	
141.	case 'FAR':
142.	\$raw_data = array(
143.	array('Scan STRA' => \$this->input-
144.	>post('checklist_scan_stra')),
145.	array('Scan Surat pernyataan mempunyai tempat
146.	praktek profesi atau surat keterangan dari pimpinan

Tabel 6.13 Implementasi algoritme menyetujui pengajuan permohonan

No	Kode Program
147.	fasilitas Pelayanan Kefarmasian' => \$this->input-
148.	>post('checklist_pernyataan_tmpt_praktek_orPim_faskes')),
149.	array('Scan Surat Persetujuan Atasan
150.	Langsung/Pemilik Sarana' => \$this->input-
151.	>post('checklist_persetujuan_atasan')),
152.	array('Scan Surat rekomendasi dari organisasi
153.	profesi' => \$this->input-
154.	>post('checklist_rekom_org_profesi')),
155.);
156.	array_push(\$data, \$raw_data);
157.	break;
158.	
159.	}
160.	
161.	\$data_to_store = array(
162.	'id_pengajuan' => \$id_pengajuan,
163.	'id_kepegawaian'=>
164.	\$SESSION['id_kepegawaian'],
165.	'id_jabatan' => \$SESSION['id_jabatan'],
166.	'diterima' => \$diterima,
167.	'info_tambahan' => \$info_tambahan,
168.	'kelengkapan' => \$grup_lengkap,
169.	'kekurangan_berkas' => \$kekurangan_berkas,
170.	'checklist_berkas' => json_encode(\$data)
171.);
172.	
173.	\$simpan = \$this->M_penanganan-
174.	>set_persetujuan(\$data_to_store);
175.	
176.	if (\$simpan[0]['status'] == 'success' \$simpan ==
177.	true) {
178.	\$this->Sys_notif->success("Persetujuan pada berkas
179.	permohonan Surat Izin berhasil berikan.");
180.	} else {
181.	\$this->Sys_notif->danger("Persetujuan pada berkas
182.	permohonan Surat Izin gagal berikan.");
183.	}
	redirect(base_url('detil-pengajuan/'.\$id_pengajuan));
	}

Tabel 6.14 Penjelasan implementasi algoritme menyetujui pengajuan permohonan

Baris	Penjelasan
1	Melakukan deklarasi <i>function</i> dengan nama kirim_persetujuan_pp dengan parameter id_pengajuan
2-8	Mengambil nilai masukan dari view dan disimpan kedalam variabel
10	Deklarasi variabel array dengan nama data
11	Memisahkan nilai variabel id_pengajuan dengan parameter "-" dan memasukkan hasilnya ke variabel raw_jenis
12	Menyimpan nilai variabel array raw_jenis index 1 kedalam variabel raw_jenis_perizinan
13	Deklarasi variabel jenis_perizinan dan mengisinya dengan nilai null
15	Deklarasi seleksi kondisi switch dengan parameter nilai variabel

Tabel 6.14 Penjelasan implementasi algoritme menyetujui pengajuan permohonan

Baris	Penjelasan
16-61	Sintaks yang akan dijalankan saat kondisi "BDN" terpenuhi
64-100	Sintaks yang akan dijalankan saat kondisi "DOK" terpenuhi
103-121	Sintaks yang akan dijalankan saat kondisi "PRW" terpenuhi
123-138	Sintaks yang akan dijalankan saat kondisi "TTK" terpenuhi
141-157	Sintaks yang akan dijalankan saat kondisi "FAR" terpenuhi
161-170	Penyimpanan nilai yang diambil dari view kedalam <i>array</i> dengan nama <i>data_to_store</i>
173-174	Memanggil fungsi <i>set_persetujuan</i> dengan memasukkan parameter <i>array data_to_store</i>
176-182	Seleksi kondisi untuk memeriksa apakah proses memasukkan berhasil atau tidak untuk menentukan pesan kembalian ke <i>view</i>
184-185	Mengarahkan ke halaman detail pengajuan

6.1.3 Implementasi algoritme membuat jadwal kunjungan

Implementasi algoritme *function* *buat_jadwal* merupakan implementasi fungsi sistem untuk menangkap jadwal kunjungan yang dibuat oleh Kepala Bidang. Tabel 6.15 merupakan implementasi algoritme dari aktivitas membuat jadwal kunjungan dan Tabel 6.16 merupakan penjelasan baris kode implementasi.

Tabel 6.15 Implementasi algoritme membuat jadwal kunjungan

No	Kode Program
1.	<code>public function buat_jadwal(\$id_pengajuan, \$tanggal_input =</code>
2.	<code>NULL, \$waktu_input = NULL, \$catatan_input = NULL)</code>
3.	<code>{</code>
4.	<code> if (\$tanggal_input == NULL && \$waktu_input == NULL &&</code>
5.	<code> \$catatan_input == NULL) {</code>
6.	<code> date_default_timezone_set("Asia/Bangkok");</code>
7.	<code> \$this->load->model('penanganan/M_jadwal');</code>
8.	<code> \$tanggal = \$this->input->post('tanggal_survei');</code>
9.	<code> \$tanggal = date("Y-m-d", strtotime(\$tanggal));</code>
10.	<code> \$waktu = \$this->input->post('jam_survei');</code>
11.	<code> \$waktu = date("H:i:s", strtotime(\$waktu));</code>
12.	<code> \$catatan = \$this->input->post('info-tambahan');</code>
13.	<code> } else {</code>
14.	<code> date_default_timezone_set("Asia/Bangkok");</code>
15.	<code> \$this->load->model('penanganan/M_jadwal');</code>
16.	<code> \$tanggal = \$tanggal_input;</code>
17.	<code> \$tanggal = date("Y-m-d", strtotime(\$tanggal));</code>
18.	<code> \$waktu = \$waktu_input;</code>
19.	<code> \$waktu = date("H:i:s", strtotime(\$waktu));</code>
20.	<code> \$catatan = \$catatan_input;</code>
21.	<code> }</code>
22.	
23.	<code> \$insert_data = \$this->M_jadwal-></code>
24.	<code> createJadwal(\$id_pengajuan, \$tanggal, \$waktu, \$catatan);</code>
25.	
26.	<code> redirect('penanganan/C_penanganan/penanganan_jadwal_kabi</code>
27.	<code> d/'.\$id_pengajuan);</code>
28.	<code>}</code>

Tabel 6.16 Penjelasan implementasi algoritme membuat jadwal kunjungan

Baris	Penjelasan
1-2	Melakukan deklarasi <i>function</i> dengan nama buat jadwal dengan parameter id_pengajuan, tanggal_input, waktu_input, catatan_input
4-5	Seleksi kondisi untuk memeriksa nilai parameter jika bernilai null
6	Deklarasi zona waktu
7	Load kelas model M_jadwal yang ada pada folder penanganan
8-12	Mengambil nilai masukan dari <i>view</i> dan disimpan kedalam variabel
13	Jika kondisi if tidak terpenuhi
14	Deklarasi zona waktu
15	Load kelas model M_jadwal yang ada pada folder penanganan
16-20	Mengambil nilai jadwal dari parameter
23-24	Memanggil fungsi createJadwal dengan memasukkan parameter nilai dari variabel tanggal, waktu dan catatan
26-27	Mengarahkan ke function penanganan_jadwal_kabid yang berada di kelas C_penanganan

6.1.4 Implementasi algoritme menugaskan pembuatan surat izin

Implementasi algoritme *function* kirim_persetujuan_kabid merupakan implementasi fungsi sistem untuk mengolah masukan dari view yang digunakan untuk mengambil dan mengirimkan masukan terkait penugasan pencetakan surat izin dari Kepala Bidang ke Petugas Proses. Tabel 6.17 berikut menunjukkan implementasi algoritme menugaskan pembuatan surat izin dan Tabel 6.18 penjelasan dari kode implementasi.

Tabel 6.17 Implementasi menugaskan pembuatan surat izin

No	Kode Program
1.	function kirim_persetujuan_kabid(\$id_pengajuan)
2.	{
3.	\$info_tambahan = \$this->input->post('info-
4.	tambahan');
5.	
6.	\$data_to_store = array(
7.	'id_pengajuan' => \$id_pengajuan,
8.	'id_kepegawaian'=>
9.	\$_SESSION['id_kepegawaian'],
10.	'id_jabatan' =>
11.	\$_SESSION['id_jabatan'],
12.	'info_tambahan' => \$info_tambahan,
13.	'need_jadwal' => FALSE
14.);
15.	\$simpan = \$this->M_penanganan->
16.	set_persetujuan(\$data_to_store);
17.	
18.	if (\$simpan[0]['status'] == 'success' \$simpan
19.	== true) {
20.	\$this->Sys_notif->
21.	success("Persetujuan diteruskan untuk pencetakan surat
22.	izin.");
23.	}else{

Tabel 6.17 Implementasi menugaskan pembuatan surat izin

No	Kode Program
24.	<code>\$this->Sys_notif-</code>
25.	<code>>danger("Persetujuan diteruskan untuk pencetakan surat izin</code>
26.	<code>gagal diberikan.");</code>
27.	<code>}</code>
28.	<code>redirect(base_url('detil-</code>
29.	<code>pengajuan/'.\$id_pengajuan));</code>
30.	<code>}</code>

Tabel 6.18 Penjelasan implementasi algoritme menugaskan pembuatan surat izin

Baris	Penjelasan
1	Melakukan deklarasi <i>function</i> dengan nama <code> kirim_persetujuan_kabid</code> dengan parameter <code>id_pengajuan</code>
3	Mengambil nilai info tambahan dari view dan memasukkannya kedalam variabel <code>info_tambahan</code>
6-13	Penyimpanan nilai kedalam <i>array</i> dengan nama <code>data_to_store</code>
15-16	Memanggil fungsi <code>set_persetujuan</code> dengan memasukkan parameter <i>array</i> <code>data_to_store</code>
18-26	Seleksi kondisi untuk memeriksa apakah proses memasukkan berhasil atau tidak untuk menentukan pesan kembalian ke <i>view</i>
28-29	Mengarahkan ke halaman detail pengajuan

6.2 Implementasi antarmuka

6.2.1 Implementasi halaman mengajukan permohonan

Tenaga Kesehatan

Rosi Cinditya T...
Perawat

Logout

DAFTAR MENU

- Beranda
- Kelengkapan Data Diri
- Pengajuan Surat Izin

Permohonan Surat Izin Kerja (SIK)

Pastikan Anda telah memenuhi seluruh berkas persyaratan yang ditentukan!

Data Diri

Pastikan bahwa **data diri** yang akan diajukan untuk permohonan surat izin telah benar! Jika ada kesalahan, **klik disini untuk merubah data**

Pas Foto

Nama Lengkap
Rosi Cinditya Tusya Putri

Tempat Lahir
Mojokerto

Tanggal Lahir (dd-mm-yyyy)
00-00-0000

Jenis Kelamin
P

Lulusan
(belum diisi)

Tahun Lulusan
(belum diisi)

Nomor SIP/STR
(belum diisi)

Alamat Rumah
Jl. Mawar no.10, Kota Malang

Tempat Kerja
(belum diisi)

Nama Sarana

Alamat Sarana

Nomor Telepon Sarana

Berkas Persyaratan

*Seluruh berkas yang akan diunggah wajib dalam format (.pdf) dan masing-masing beres kurang dari 2MB

Scan Kartu Tanda Penduduk (KTP) yang masih berlaku No file chosen

Scan SIP/STR yang masih berlaku No file chosen

Scan surat keterangan sehat dari dokter yang memiliki SIP No file chosen

Scan surat Keterangan dari pimpinan sarana pelayanan kesehatan No file chosen

Scan Surat Rekomendasi dari Organisasi Profesi No file chosen

Copyright © 2017-2018 Sistem Pengajuan Izin Kerja dan Praktik Tenaga Kesehatan Dinas Kesehatan Kota XYZ. All rights reserved. Version 1.0.0

Gambar 6.1 Implementasi halaman mengajukan permohonan

Gambar 6.1 merupakan implementasi antarmuka dari halaman mengajukan permohonan bagi Tenaga Kesehatan kelompok atau jenis Perawat. Secara garis besar halaman mengajukan permohonan untuk seluruh kelompok Tenaga Kesehatan memiliki tampilan yang sama, namun isi formulir dan berkas persyaratan saja yang membedakan, bergantung pada ketentuan yang telah ditentukan oleh pihak Dinkes. Halaman ini berisi formulir pengisian data tempat kerja atau praktik dan formulir unggah berkas persyaratan dalam ekstensi (.pdf). Ketentuan terkait jumlah dan macam-macam data yang perlu dimasukkan oleh Tenaga Kesehatan pada halaman ini, mengacu pada dokumen *template* pengajuan permohonan izin praktik/kerja yang didapatkan dari Dinkes Kota XYZ melalui CV. Sarana Utama Solusindo.

6.2.2 Implementasi halaman menyetujui pengajuan permohonan

AdminSINKES

Bambang Susilo
Online
Logout

MENU NAVIGASI

- Dashboard
- Berkas Pemohon
- Semua permohonan
- Sedang ditangani
- Jadwal Survei

AKUN

Ganti Password

Detail Pengajuan #SIP-PRW-180617053624552

Note:
Halaman ini berisi detail pengajuan dari ID - SIP-PRW-180617053624552 yang berisi data diri Tenaga Kesehatan sebagai pemohon dan berkas persyaratan yang dilampirkan.

Riwayat Penanganan

Show 10 entries Search:

Tanggal	Waktu	Petugas	Status	Keterangan
17-06-2018	10:36:24	-	Masuk	Berkas permohonan telah dikirimkan ke Dinas Kesehatan Kota XYZ

Showing 1 to 1 of 1 entries Previous 1 Next

Tindakan :

Checklist Kelengkapan Berkas

Nama Berkas	Status		
	Benar	Salah	Tidak Disertakan
Scan Kartu Tanda Penduduk (KTP) yang masih berlaku	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Scan SIP/STR yang masih berlaku	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Scan surat keterangan sehat dari dokter yang memiliki SIP	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Scan surat Keterangan dari pimpinan sarana pelayanan kesehatan	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Scan Surat Rekomendasi dari Organisasi Profesi	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Pilih tindakan terhadap permohonan yang diajukan :
- Pilih -

Catatan :
(belum diisi)

Kirim

SIP-PRW-180617053624552 Diajukan pada : 17-06-2018 | Pukul 10:36:24

Data Diri Pemohon

Pas Foto:

Nama Lengkap: Danil Reza Pridausy
Tempat/Tanggal Lahir: Surabaya, 26 Februari 1996
Jenis Kelamin: Laki-laki
Asal Lulusan:
Tahun Lulusan: 0000
Nomor SIP/STR: -
Alamat Rumah: Bimbing Indah Megah Blok J no 9, Kota Malang
Tempat Kerja: Jl. Kenangan

Berkas Pendukung

Nama Sarana Tempat Praktik: Sarana baru
Alamat Sarana Tempat Praktik: Jl. saaran
Nomor Telepon Sarana Tempat Praktik: 20203030
Scan Kartu Tanda Penduduk (KTP) yang masih berlaku: [Lihat](#)
Scan SIP/STR yang masih berlaku: [Lihat](#)
Scan surat keterangan sehat dari dokter yang memiliki SIP: [Lihat](#)
Scan surat Keterangan dari pimpinan sarana pelayanan kesehatan: [Lihat](#) *data tidak dilampirkan.
Scan Surat Rekomendasi dari Organisasi Profesi: [Lihat](#)

Copyright © 2017-2018 Sistem Pengajuan Izin Kerja dan Praktik Tenaga Kesehatan Dinas Kesehatan Kota XYZ. All rights reserved. Version 1.0.0

Gambar 6.2 Implementasi halaman menyetujui pengajuan permohonan

Gambar 6.2 merupakan implementasi antarmuka dari halaman menyetujui atau menolak permohonan yang telah diajukan oleh Tenaga Kesehatan. Jika data permohonan dan berkas yang dilampirkan sesuai dengan persyaratan yang ditentukan, maka Petugas Proses dapat menyetujui penanganan selanjutnya. Namun jika berkas yang dilampirkan dinilai tidak sesuai, maka Petugas Proses dapat memilih untuk menolak permohonan dan penanganan terhadap laporan tersebut tidak akan diteruskan. Setelah pengisian tindakan selesai, formulir

pengisian tindakan akan hilang secara otomatis. Halaman ini terdiri dari tabel berisi informasi *log* perubahan status permohonan, formulir pengisian tindakan dan detail permohonan.

6.2.3 Implementasi halaman membuat jadwal kunjungan

AdminSINKES

Susanti Ambarwati
Online

Logout

MENU NAVIGASI

- Dashboard
- Berkas Pemohon
- Semua permohonan
- Sedang ditangani
- Jadwal Survei
- AKUN
- Ganti Password

Detail Pengajuan #SIP-BDN-180503193502679

Home - Detail Pengajuan

Note:
Halaman ini berisi detail pengajuan dari ID : **SIP-BDN-180503193502679** yang berisi data diri Tenaga Kesehatan sebagai pemohon dan berkas persyaratan yang dilampirkan.

Riwayat Penanganan

Show 10 entries Search:

Tanggal	Waktu	Petugas	Status	Keterangan
04-05-2018	00:35:02	-	Menolak	Berkas permohonan telah dikirimkan ke Dinas Kesehatan Kota XYZ
05-05-2018	16:16:51	Bambang Susilo	Diterima Petugas Prorot	sadsadsdd
06-05-2018	09:42:45	Handoko Setiabudi	Disetujui Kepala Seksi BDM	Lanjut pada penjadwalan survei

Showing 1 to 3 of 3 entries

Previous 1 Next

Buat Jadwal Kunjungan Survei :

Tanggal: mm/dd/yyyy

Jam: 07:45 AM

Catatan : (belum diisi)

Kirim

Jadwal Visitasi

SIP-BDN-180503193502679

Diajukan pada : 04-05-2018 | Pukul 00:35:02

Data Diri Pemohon

Berkas Pendukung

Copyright © 2017-2018 Sistem Pengajuan Izin Kerja dan Praktik Tenaga Kesehatan Dinas Kesehatan Kota XYZ. All rights reserved. Version 1.0.0

Gambar 6.3 Implementasi halaman membuat jadwal kunjungan

Gambar 6.3 merupakan implementasi antarmuka dari halaman membuat jadwal kunjungan survei ke tempat praktik Tenaga Kesehatan. Halaman ini terdiri dari tabel berisi informasi *log* perubahan status permohonan, formulir pembuatan jadwal dan detail permohonan. Bagian formulir pembuatan jadwal akan hilang secara otomatis jika jadwal telah selesai dibuat oleh Kepala Bidang, namun akan dapat muncul kembali ketika berita acara kunjungan dinyatakan tidak lolos, sehingga mengharuskan penjadwalan ulang kunjungan.

6.2.4 Implementasi halaman menugaskan pembuatan surat izin

AdminSINKES

Susanti Ambarwati
Online

Logout

MENU NAVIGASI

- Dashboard
- Berkas Pemohon
- Semua permohonan
- Sedang ditangani
- Jadwal Survei
- AKUN
- Ganti Password

Detail Pengajuan #SIP-BDN-180522162257551

Home -> Detail Pengajuan

Note:
Halaman ini berisi detail pengajuan dari ID : **SIP-BDN-180522162257551** yang berisi data diri Tenaga Kesehatan sebagai pemohon dan berkas persyaratan yang dilampirkan.

Riwayat Penanganan

Show 10 entries

Tanggal	Waktu	Petugas	Status	Keterangan
22-05-2018	21:24:44	Susanti Ambarwati	Jadwal kunjungan survei telah dibuat	Jadwal telah dibuat Kepala Bidang SDM
22-05-2018	21:24:14	Handoko Setiabudi	Diterima Kepala Seksi SDM	lanjut
22-05-2018	21:23:12	Bambang Susilo	Diterima Petugas Proses	lengkap
22-05-2018	21:22:57	-	Masuk	Berkas permohonan telah dikirimkan ke Dinas Kesehatan Kota XYZ
20-05-2018	07:19:57	Handoko Setiabudi	Penilaian berita acara	Berkas masuk dalam tahap penilaian KABID

Showing 1 to 5 of 5 entries

Previous 1 Next

Hasil Visitasi Lapangan :

Hasil

SDM Pendukung

- Tenaga Kesehatan lain
- Tenaga Non Kesehatan

Bangunan dan Ruang

- Bangunan
- Ruang tunggu

Prasarana

- Sistem Air bersih
- Ventilasi atau sirkulasi Udara
- Prasarana lain sesuai kebutuhan
- Kursi Kerja
- Lemari Arsip
- Mesa Tulis 1/2 biro
- Buku KIA
- Bagan Dinding MTBS
- Formulir Persalinan/Infas dan KB
- Formulir Partograf

Peralatan

- Bak instrumen dengan tutup
- Stetoskop Dewasa (2)
- Bantal

Obat dan Bahan Habis Pakai

- Oksitosin inj 10 IU /mL
- Kalsium glukonat inj 10%
- Kombinasi desogestrel dan etinilestradiol Tablet
- Infus Set dengan Wing Needle untuk Anak dan Bayi nomor 23 dan 25

Standar Prosedur Operasional

- SPO Pelayanan Antenatal
- SPO Penanganan Bayi Asfiksia
- SPO Mengatasi Syok

Hasil Peninjauan :

Tinjau

Hasil Kesimpulan :

Kesimpulan

Hasil Penilaian Berkas

Penilaian Kemajuan

Pilih hasil penilaian

Kirim

SIP-BDN-180522162257551

Ditajukan pada : 22-05-2018 | Pukul 21:22:57

Data Diri Pemohon + Berkas Pendukung +

Copyright © 2017-2018 Sistem Pengajuan Izin Kerja dan Praktik Tenaga Kesehatan Dinas Kesehatan Kota XYZ. All rights reserved. Version 1.0.0

Gambar 6.4 Implementasi halaman menugaskan pembuatan surat izin

Gambar 6.4 merupakan implementasi antarmuka dari halaman menugaskan pembuatan surat izin ketika berita acara kunjungan dinyatakan lolos persyaratan oleh Kepala Bidang . Halaman ini terdiri dari tabel berisi informasi *log* perubahan status permohonan, formulir penugasan dan detail permohonan. Penugasan yang dikirimkan oleh Kepala Bidang berfungsi sebagai penanda bahwa Petugas Proses telah diperbolehkan mencetak surat izin dan menyelesaikan proses selanjutnya.

BAB 7 PENGUJIAN

Pengimplementasian *V-Model* pada penelitian ini difokuskan pada tahap *Requirements Analysis, Specifications, Architectural Design, Detail Design*, dan *Coding*. Sedangkan pada fase pengujian berfokus pada pengujian penerimaan (*acceptance testing*) dikarenakan sebelum pengujian penerimaan dilaksanakan, sistem telah dipastikan berjalan sebagaimana mestinya dari unit terkecil hingga satu kesatuan fungsional. Maka dari itu pengujian unit (*unit testing*), pengujian integrasi (*integration testing*) dan pengujian sistem (*system testing*) tidak dibahas secara khusus dan mendalam.

7.1 Perancangan pengujian sistem

Sebelum melakukan pengujian fungsional maupun non fungsional terhadap SINKES, maka pada bagian ini dijabarkan mengenai rencana pengujian sebagai acuan dalam pengujian yang akan dilakukan. Berikut adalah rencana pengujian yang dibagi berdasarkan masing-masing jenis pengujian yang akan dilakukan.

7.1.1 Responsiveness testing

Pada pengujian *responsiveness* ini, pengujian akan dilakukan melalui website <http://responsivedesignchecker.com/> sebagai alat bantu dalam memeriksa tingkat *responsiveness* yang telah dibuat. Berikut urutan pengujian yang akan dilakukan.

1. Mengakses website <http://responsivedesignchecker.com/>.
2. Memasukkan *link* SINKES kedalam formulir pengisian yang tersedia.
3. Mendapatkan hasil berupa tampilan pada berbagai macam perangkat yang masing-masing memiliki resolusi berbeda.

Berikut pada Tabel 7.1 ditunjukkan kasus uji yang akan dilakukan. Kolom Jenis Perangkat berikan jenis perangkat yang diujikan, kemudian Ukuran Resolusi berisikan nilai dari resolusi perangkat tersebut. Jenis perangkat yang dipilih berdasarkan 3 perangkat yang umum digunakan yaitu perangkat jenis *desktop*, tablet dan telepon genggam. Kolom Hasil berisikan hasil tampilan dari sistem pada jenis resolusi yang diajukan, sedangkan pada kolom Status berisikan keterangan apakah sistem dapat beradaptasi dengan ukuran resolusi atau tidak.

Tabel 7.1 Rancangan kasus Uji *responsiveness testing*

No.	Jenis Perangkat	Ukuran Resolusi	Hasil	Status
1	Desktop	1440*900		
2	Tablet	768*1024		
3	Telepon genggam	414*736		

7.1.2 Compatibility testing

Setelah melakukan pengujian *responsiveness*, maka pengujian yang dilakukan selanjutnya adalah pengujian *compatibility*. Seperti yang telah dijelaskan pada Bab 2 poin 2.11.2 alat bantu yang digunakan dalam pengujian ini adalah aplikasi perangkat lunak *Sortsite* versi *desktop* yang akan memeriksa komponen *tag* HTML, CSS, *script*, kesalahan dalam pemanggilan halaman, teknologi yang tidak didukung oleh *browser* tertentu. Berikut adalah rencana pengujian yang akan dilakukan.

1. Membuka aplikasi *Sortsite* versi *desktop*.
2. Memasukkan *link* SINKES kedalam *address bar* yang tersedia.
3. Memilih tombol “*check*”.
4. Setelah keluar hasil pengujian yang dilakukan oleh *Sortsite*, pilih fungsi untuk melihat hasil pengujian *compability*.
5. Hasil ditampilkan berupa gambar dan keterangan dari setiap *error* yang ditemukan.

Berikut pada Tabel 7.2 merupakan perencanaan kasus uji yang akan dijadikan dasar pengujian.

Tabel 7.2 Rancangan kasus uji *compatibility testing*

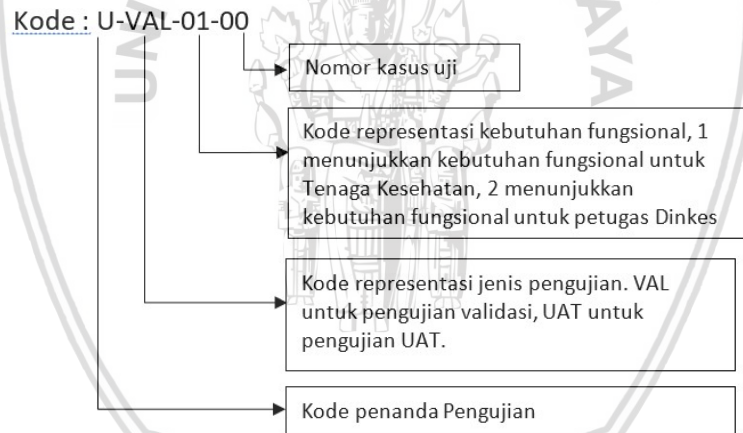
No.	Jenis Browser	Versi	Hasil	Status		
				Critical issues	Major issues	Minor issues
1	Internet Explorer	9				
		10				
		11				
2	Edge	16				
3	Firefox	60				
4	Safari	≤ 10				
		11				
5	Opera	51				
6	Chrome	66				
7	iOS	≤ 9				
		10				
		11				
8	Android	≤ 3				
		4				
9	BlackBerry	≤ 7.1				

Tabel 7.2 Rancangan kasus uji *compatibility testing*

No.	Jenis Browser	Versi	Hasil	Status		
				Critical issues	Major issues	Minor issues
		10				

7.1.3 User Acceptance Testing (UAT)

Pengujian UAT dilaksanakan dengan menggunakan 2 pengujian, yaitu pengujian validasi untuk memastikan bahwa setiap nilai masukan pengguna dapat menghasilkan keluaran yang sesuai dengan yang seharusnya dan pengujian penerimaan pengguna terhadap sistem. pengujian validasi dilakukan sendiri oleh penulis selaku pengembang sistem, sedangkan pengujian penerimaan dilakukan oleh perwakilan CV. Sarana Utama Solusindo selaku pihak yang bertanggungjawab untuk memastikan bahwa sistem yang dibuat telah sesuai dengan harapan dari pihak Dinkes Kota XYZ. Dalam melakukan pengujian ini, terdapat pemberian kode sebagai penanda kasus uji yang dirancang. Ketentuan pemberian kode ditunjukkan dalam Gambar 5.15.



Gambar 7.1 Pengkodean untuk menandai kasus uji pengujian UAT

7.1.3.1 Mekanisme UAT untuk validasi sistem

Tabel 5.13 berikut merupakan desain kasus uji UAT validasi fungsionalitas dari aktor Tenaga Kesehatan. Kolom kode pengujian berisi kode penanda kasus uji. Kolom kode *use case* berisi kode dari *use case* yang diuji, ditulis untuk memudahkan peruntutan dengan dokumentasi perancangan yang telah dibuat. Sedangkan Tabel 5.14 merupakan desain kasus uji UAT validasi fungsionalitas aktor dari pihak Dinkes yaitu Petugas Proses; Kepala Seksi Kefarmasian, Kepala Seksi SDM; UPT Puskesmas Wilayah, Seksi Kesehatan Keluarga dan Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas.

Tabel 7.3 Rancangan kasus uji UAT validasi aktor Tenaga Kesehatan

Kode Pengujian	Kode <i>Use Case</i>	Fungsional Sistem	Kasus Uji	Yang Diharapkan	Hasil (Valid/Tidak valid)
U-VAL-01-01	UCT-01	Membuat Akun	Pada halaman pertama sistem, penguji memilih fungsi mendaftarkan diri, kemudian memasukkan data diri seperti nama lengkap, Nomor Induk Kependudukan (NIK), <i>email</i> , nomor <i>handphone</i> , <i>password</i> dan jenis kelompok Tenaga Kesehatan. Setelah mengisi, centang pernyataan kebenaran data. Kemudian pilih tombol Daftar.	Penguji dapat berhasil membuat akun dan masuk ke halaman beranda untuk Tenaga Kesehatan.	
			Pada halaman pertama sistem, penguji memilih fungsi mendaftarkan diri, kemudian memasukkan data diri dengan email yang pernah didaftarkan sebelumnya.	Penguji mendapatkan pesan gagal registrasi karena email telah terdaftar.	
U-VAL-01-02	UCT-02	<i>Login</i>	Pada halaman utama sistem, penguji memasukkan Nomor Induk Kependudukan (NIK) dan <i>password</i> dengan hak akses Tenaga Kesehatan, lalu menekan tombol <i>Login</i>	Penguji dapat masuk ke halaman beranda untuk Tenaga Kesehatan.	
			Pada halaman utama sistem, penguji memasukkan Nomor Induk	Penguji mendapatkan pesan gagal <i>login</i> karena NIK dan <i>password</i> tidak sesuai.	

Tabel 7.3 Rancangan kasus uji UAT validasi aktor Tenaga Kesehatan

Kode Pengujian	Kode <i>Use Case</i>	Fungsional Sistem	Kasus Uji	Yang Diharapkan	Hasil (Valid/Tidak valid)
			Kependudukan (NIK) dan <i>password</i> yang tidak sesuai.		
U-VAL-01-03	UCT-03	Merubah Biodata	Pada halaman berisi formulir perubahan biodata, penguji mengisi sesuai data yang ingin dirubah. Setelah itu pilih tombol simpan.	Penguji mendapatkan pesan berhasil merubah biodata.	
			Perubahan yang dilakukan gagal disimpan oleh sistem.	Penguji mendapatkan pesan gagal merubah biodata.	
U-VAL-01-04	UCT-09	Merubah Foto	Pada halaman berisi formulir perubahan biodata, penguji memilih tombol ganti foto. Kemudian tekan tombol untuk memilih <i>file</i> . Jika foto telah dipilih, tekan tombol simpan.	Penguji mendapatkan pesan berhasil merubah foto dan kembali ke halaman ubah biodata.	
U-VAL-01-05	UCT-04	Membuat Berkas Pendukung	Penguji memilih menu Berkas Pendukung. Setelah formulir pembuatan berkas ditampilkan, isikan data yang diperlukan sesuai dengan jenis berkas yang akan dibuat, lalu pilih tombol Cetak.	Penguji mendapatkan berkas pendukung yang telah terisi.	
U-VAL-01-06	UCT-05	Mengajukan Permohonan	Pada halaman berisi formulir pengajuan permohonan, penguji mengisikan data dan berkas persyaratan yang sesuai dengan masing-masing jenis kelompok	Penguji mendapatkan pesan berhasil mengirimkan permohonan surat izin kerja/praktik.	

Tabel 7.3 Rancangan kasus uji UAT validasi aktor Tenaga Kesehatan

Kode Pengujian	Kode <i>Use Case</i>	Fungsional Sistem	Kasus Uji	Yang Diharapkan	Hasil (Valid/Tidak valid)
			Tenaga Kesehatan. Setelah mengisi, tekan tombol Ajukan Permohonan.		
U-VAL-01-07	UCT-06	Melihat Alur Penanganan	Penguji memilih tombol Beranda.	Penguji berada pada halaman beranda yang berisi tabel <i>log</i> penanganan permohonan.	
U-VAL-01-08	UCT-07	Mencetak Tanda Bukti	Pada tabel berisi <i>log</i> status permohonan yang diajukan, penguji memilih tombol Cetak Tanda Bukti jika permohonan telah disetujui petugas Dinkes.	Penguji mendapatkan tanda bukti terima berkas.	
U-VAL-01-09	UCT-08	<i>Logout</i>	Penguji memilih tombol <i>Logout</i> .	Penguji kembali ke halaman <i>login</i> .	

Tabel 7.4 Rancangan kasus uji UAT validasi fungsional aktor Petugas Proses; Kepala Seksi Kefarmasian, Kepala Seksi SDM; UPT Puskesmas Wilayah, Seksi Kesehatan Keluarga dan Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang Diharapkan	Hasil (Valid/Tidak valid)
U-VAL-02-01	UCP-01	Login	Pada halaman utama sistem, penguji memasukkan <i>username</i> dan <i>password</i> , lalu menekan tombol <i>Login</i> .	Penguji dapat berhasil membuat akun dan masuk ke halaman beranda untuk petugas Dinkes.	
			Pada halaman utama sistem, penguji memasukkan <i>username</i> dan <i>password</i> yang tidak sesuai, lalu menekan tombol <i>Login</i> .	Penguji mendapatkan pesan gagal <i>login</i> karena <i>username</i> dan <i>password</i> tidak sesuai.	
U-VAL-02-02	UCP-02	Menyetujui Pengajuan Permohonan	Pada halaman Detil Permohonan, penguji memilih menerima atau tidak, mengisi catatan tambahan, memilih lengkap/tidaknya berkas, mengisi kekurangan berkas bila ada, kemudian pilih tombol Kirim.	Penguji mendapatkan pesan berhasil mengirimkan persetujuan pengurusan permohonan.	
U-VAL-02-03	UCP-03	Menyetujui Penanganan Lebih Lanjut	Pada halaman Detil Permohonan, penguji mengisi catatan tambahan, kemudian pilih tombol Kirim.	Penguji mendapatkan pesan berhasil mengirimkan persetujuan untuk penanganan lebih lanjut.	
U-VAL-02-04	UCP-04	Membuat Jadwal Kunjungan	Pada halaman Detil Permohonan, penguji memasukkan tanggal dan jam kunjungan pada formulir buat jadwal, kemudian pilih tombol Simpan.	Penguji mendapatkan pesan berhasil membuat jadwal kunjungan visitasi.	

Tabel 7.4 Rancangan kasus uji UAT validasi fungsional aktor Petugas Proses; Kepala Seksi Kefarmasian, Kepala Seksi SDM; UPT Puskesmas Wilayah, Seksi Kesehatan Keluarga dan Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas

Kode Pengujian	Kode <i>Use Case</i>	Fungsional Sistem	Kasus Uji	Yang Diharapkan	Hasil (Valid/Tidak valid)
U-VAL-02-05	UCP-05	Melihat Daftar Jadwal Kunjungan	Penguji memilih tombol Jadwal Kunjungan.	Penguji berhasil melihat daftar jadwal kunjungan yang telah dibuat.	
U-VAL-02-06	UCP-06	Membuat Berita Acara	Pada halaman Detil Permohonan, penguji mengisi setiap poin yang ada pada formulir, kemudian pilih tombol Kirim.	Penguji mendapatkan pesan berhasil membuat berita acara.	
U-VAL-02-07	UCP-07	Menugaskan Pembuatan Surat Izin	Pada halaman Detil Permohonan, penguji mengisi catatan tambahan, kemudian pilih tombol Kirim.	Penguji mendapatkan pesan berhasil mengirimkan penugasan pencetakan surat izin.	
U-VAL-02-08	UCP-08	Mencetak Surat Izin	Pada halaman detil Permohonan, penguji memilih tombol Cetak Surat.	Penguji mendapatkan surat izin kerja/praktik yang telah terisi oleh sistem.	
U-VAL-02-09	UCP-09	Pengujian <i>Logout</i>	Penguji memilih tombol <i>Logout</i> .	Penguji kembali ke halaman <i>login</i> petugas Dinkes.	

7.1.3.2 Mekanisme UAT dampak setelah adanya sistem

Pengujian dampak adanya sistem dilakukan dengan cara mendemonstrasikan program kepada responden yang terdiri dari Tenaga Kesehatan dan perwakilan CV. Sarana Utama Solusindo sebagai perwakilan pihak Dinkes. Tenaga Kesehatan yang melakukan pengujian diwakilkan dari kelompok Apoteker, Dokter, Perawat dan Tenaga Teknis Kefarmasian. Kelompok Bidan tidak dilibatkan dalam pengujian karena secara garis besar, proses bisnis pengajuan permohonannya sama dengan pengajuan permohonan untuk kelompok Apoteker. Sedangkan dari CV. Sarana Utama Solusindo pada pengujian ini diwakili oleh 3 pegawai yang masing-masing memahami proses bisnis yang sedang berjalan dan permasalahan yang terjadi.

Adapun kriteria penerimaan yang akan diujikan adalah kriteria *performance*, *usability*, *functional correctness and completeness*, *confidentially and availability*. Berikut adalah uji kasus untuk validasi sistem berdasarkan fungsionalitas yang dimiliki tiap aktor dibedakan menjadi 2 tabel yaitu fungsionalitas untuk Tenaga Kesehatan dan pihak Dinkes. Tabel tersebut terdiri dari kolom nomor yang berisi nomor kriteria pengujian, Kriteria yang berisi kriteria penerimaan yang akan diujikan, Kode Kasus Uji yang berisi kode penanda kasus uji, dan Daftar Pertanyaan berisi pertanyaan-pertanyaan yang akan diajukan pada kuesioner penilaian hasil pengujian. Rancangan kasus uji UAT fungsionalitas untuk Tenaga Kesehatan ditunjukkan oleh Tabel 5.15, sedangkan untuk pihak Dinkes ditunjukkan oleh Tabel 5.16.

Tabel 7.5 Rancangan kasus uji UAT pengajuan permohonan

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
1.	<i>Performance</i>	U-UAT-01-01	1. Apakah SINKES dapat diakses dengan cepat?
2.	<i>Usability</i>	U-UAT-01-02	1. Apakah SINKES mudah digunakan?
		U-UAT-01-03	2. Apakah tampilan SINKES ketika digunakan?
		U-UAT-01-04	3. Apakah SINKES mempermudah penyampaian informasi penanganan antar petugas dan Kepala Dinas?
3.	<i>Functional Correctness and Completeness</i>	U-UAT-01-05	1. Apakah dengan adanya SINKES membantu dalam proses pengajuan permohonan surat izin kerja atau praktik?
		U-UA6-01-06	2. Apakah dengan adanya SINKES membantu dalam mengetahui penanganan permohonan yang telah diajukan?

Tabel 7.5 Rancangan kasus uji UAT pengajuan permohonan

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
4.	<i>Confidentially and Availability</i>	U-UAT-01-07	1. Apakah login user dibutuhkan untuk membantu keamanan informasi pemohon dalam SINKES?

Tabel 7.6 Rancangan kasus uji UAT penanganan permohonan

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
1.	<i>Performance</i>	U-UAT-02-01	1. Apakah SINKES dapat diakses dengan cepat?
2.	<i>Usability</i>	U-UAT-02-02	1. Apakah SINKES mudah digunakan?
		U-UAT-02-03	2. Apakah tampilan SINKES nyaman ketika digunakan?
		U-UAT-02-04	3. Apakah SINKES mempermudah penyampaian informasi penanganan antar petugas dan Kepala Dinas?
3.	<i>Functional Correctness and Completeness</i>	U-UAT-02-05	1. Apakah dengan adanya SINKES membantu dalam pencatatan permohonan masuk?
		U-UAT-02-06	2. Apakah dengan adanya SINKES membantu pemeriksaan berkas pemohon?
		U-UAT-02-07	3. Apakah dengan adanya SINKES membantu dalam memberikan pemberitahuan kepada Tenaga Kesehatan?
		U-UAT-02-08	4. Apakah dengan adanya SINKES membantu dalam melakukan tindakan penanganan?
		U-UAT-02-09	5. Apakah dengan adanya sistem dapat membantu dalam membuat jadwal kunjungan survei?
		U-UAT-02-10	6. Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan dapat memudahkan melihat jadwal kunjungan survei dan Tenaga Kesehatan yang bersangkutan?
		U-UAT-02-11	7. Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan dapat membantu pembuatan berita acara kunjungan?

Tabel 7.6 Rancangan kasus uji UAT penanganan permohonan

No	Kriteria	Kode Kasus Uji	Daftar Pertanyaan
		U-UAT-02-12	8. Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan dapat membantu penugasan pembuatan surat izin kerja/praktik?
		U-UAT-02-13	9. Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan dapat membantu mengetahui penanganan yang telah diberikan pada masing-masing tahap?
		U-UAT-02-14	10. Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan membantu dalam pengarsipan dokumen permohonan
4.	<i>Confidentially and Availability</i>	U-UAT-02-15	11. Apakah login user dibutuhkan untuk membantu keamanan informasi petugas dalam SINKES?

7.6 Hasil pengujian

Setelah mengimplementasikan rencana pengujian, maka pada bagian ini akan diuraikan mengenai hasil yang didapatkan dari pengujian yang dilakukan.

7.6.1 Hasil *responsiveness testing*

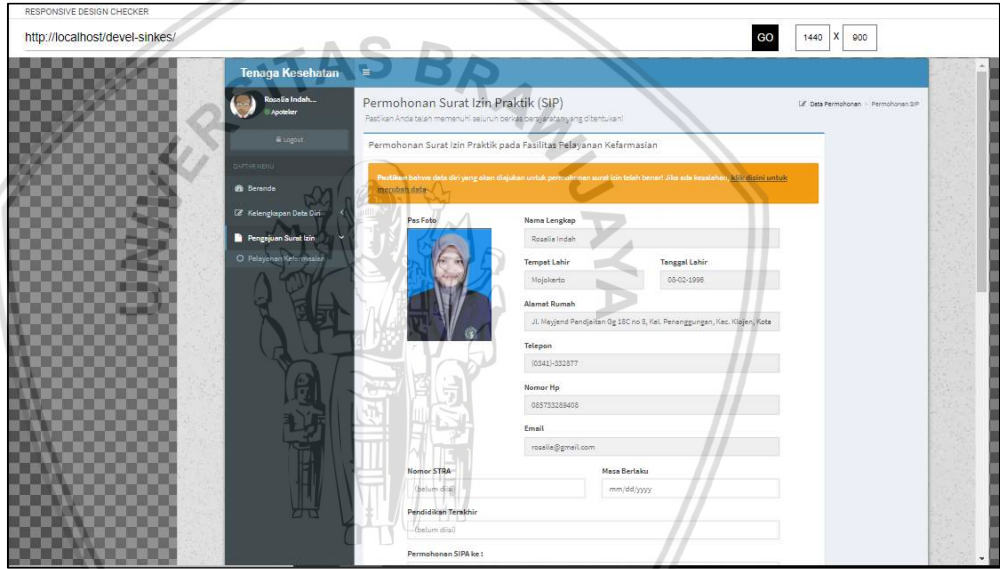
Bagian ini adalah hasil dari pengujian *responsiveness (responsiveness testing)* yang dilakukan pada website <http://responsivedesignchecker.com/> dengan menggunakan 3 ukuran resolusi layar perangkat yang berbeda, yaitu layar perangkat *desktop* (1440*900), tablet (768*1024) dan telepon genggam (414*736). Pengujian dilakukan pada halaman pengajuan permohonan untuk memastikan bahwa SINKES dapat ditampilkan dengan baik di berbagai jenis perangkat yang dimiliki oleh Tenaga Kesehatan. Berikut pada Tabel 7.13 merupakan hasil dari pengujian yang dilakukan.

Dari hasil *responsiveness testing* yang ditunjukkan pada Tabel 7.1, dapat disimpulkan bahwa pada perangkat berjenis *desktop* dengan resolusi 1440*900, sistem ditampilkan dengan memperlihatkan *background* berwarna abu-abu di kanan dan kiri layar. Selanjutnya bagian menu disebelah kiri diperlihatkan dengan jelas hingga tiap pilihan menu terlihat. Kemudian pada bagian tengah yang berisi formulir pengajuan permohonan, foto pemohon ditampilkan berdampingan

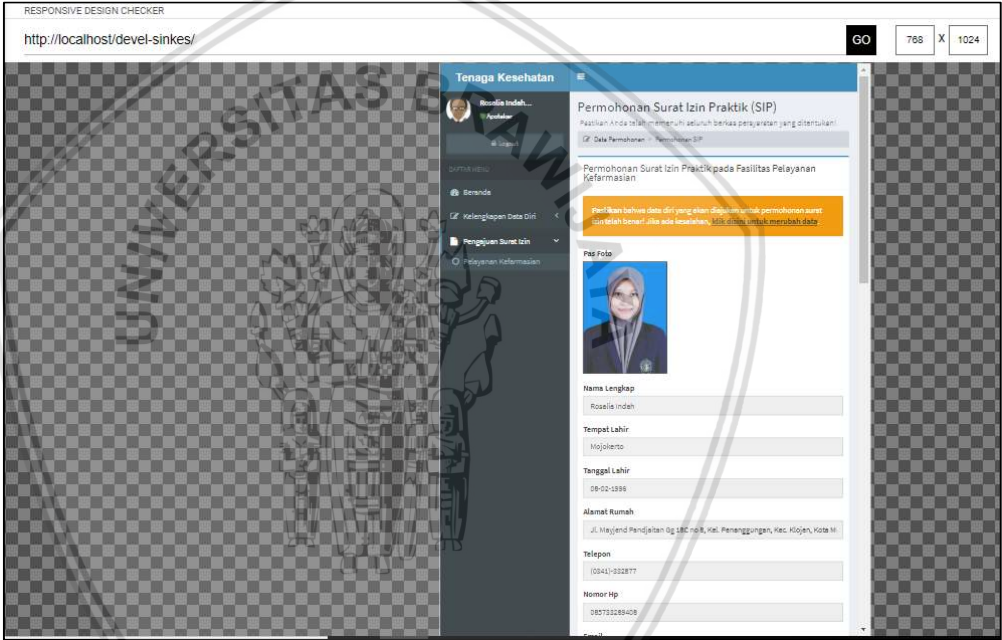
dengan kolom formulir. Sedangkan pada jenis perangkat tablet dengan ukuran resolusi 768*1024, sistem ditampilkan tanpa background abu-abu disamping kanan-kiri, namun bagian menu disamping kiri masih ditampilkan secara penuh. Untuk tampilan formulir pengajuan, terlihat perbedaan berupa foto kolom formulir yang ditampilkan dibawah foto pemohon. Pada perangkat yang terakhir yaitu telepon genggam dengan resolusi 414*736, tampilan sistem beradaptasi dengan tidak menampilkan pilihan menu seperti pada kedua perangkat sebelumnya, yaitu dengan merubahnya menjadi tombol *toggle* disebelah kiri atas layar. Kemudian pada tampilan formulir pengajuan, formulir ditampilkan seperti pada perangkat tablet.



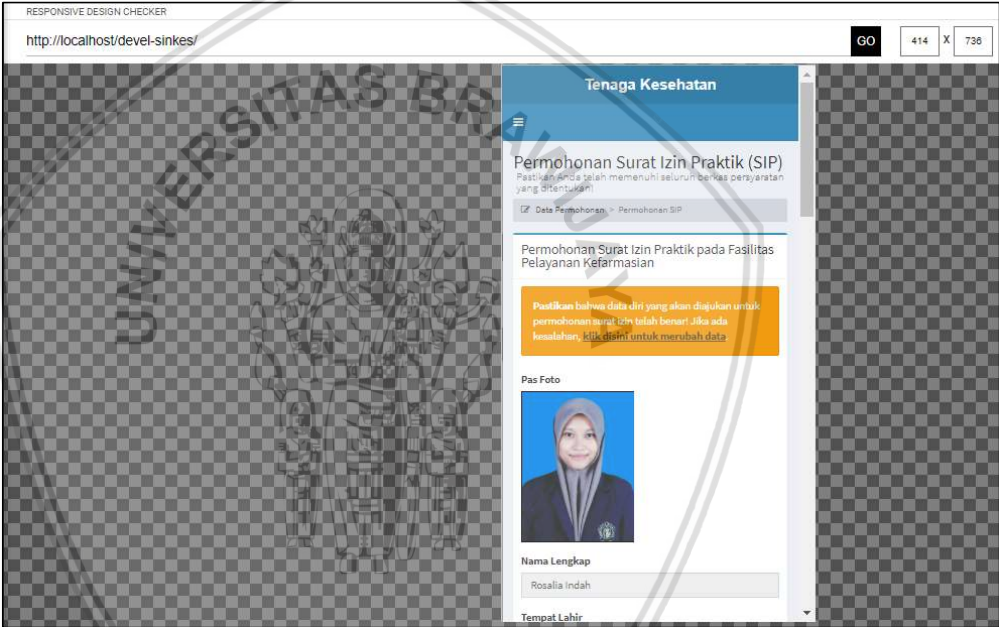
Tabel 7.7 Hasil *responsiveness testing*

No.	Jenis Perangkat	Ukuran Resolusi	Hasil	Status
1	Desktop	1440*900	 <p>Gambar 7.2 hasil <i>responsiveness testing</i> halaman pengajuan permohonan pada resolusi layar 1440*900</p>	Beradaptasi

Tabel 7.7 Hasil *responsiveness testing*

No.	Jenis Perangkat	Ukuran Resolusi	Hasil	Status
2	Tablet	768*1024	 <p>Gambar 7.3 hasil <i>responsiveness testing</i> halaman pengajuan permohonan pada resolusi layar 768*1024</p>	Beradaptasi

Tabel 7.7 Hasil *responsiveness testing*

No.	Jenis Perangkat	Ukuran Resolusi	Hasil	Status
3	Telepon genggam	414*736	 <p>Gambar 7.4 hasil <i>responsiveness testing</i> halaman pengajuan permohonan pada resolusi layar 414*736</p>	Beradaptasi

7.6.2 Hasil *compatibility testing*

Browser Version	IE 9 10 11	Edge 16	Firefox 60	Safari ≤ 10 11	Opera 51	Chrome 66	iOS ≤ 9 10 11	Android ≤ 3 4*	BlackBerry ≤ 7.1 10.0	Key
Critical Issues	✓ ✓ ✓	✓	●	✓ ✓	✓	●	✓ ✓ ✓	✓ ✓	✓ ✓	● Missing content or functionality
Major Issues	● ● ●	✓		✓ ✓	✓		✓ ✓ ✓	✓ ✓	✓ ✓	● Major layout or performance problems
Minor Issues	●	✓	●	✓ ✓	✓		✓ ✓ ✓	✓ ✓	● ✓	● Minor layout or performance problems



* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Gambar 7.5 Hasil *compatibility testing*



Pada hasil pengujian yang ditunjukkan dalam Gambar 7.4, diketahui bahwa terdapat *Critical Issues* pada jenis browser *Firefox* 60 dan *Chrome* 66. Critical issues tersebut muncul akibat tidak diaktifkannya fitur ssl atau https saat melakukan pengujian, lebih jelasnya *critical issues* terjadi pada halaman *login* untuk Tenaga Kesehatan maupun petugas dari Dinkes. Kemudian terdapat *Major Issues* yang terjadi pada jenis browser *IE (Internet Explorer)* versi 9, 10 dan 11 dikarenakan terdapat CSS yang tidak lagi didukung. Sedangkan *Minor Issues* terjadi pada browser *IE (Internet Explorer)* versi 9, *Firefox* 60 dan *BlackBerry* ≤ 7 akibat beberapa *class* CSS tidak dapat bekerja.

Berikut pada Tabel 7.15 adalah hasil yang didapatkan dari pengujian *compatibility (compatibility testing)*. Tabel terdiri jenis browser, versi dari tiap browser tersebut, hasil yang didapatkan, kemudian status hasil pengujian dengan kategori: *Critical issues*, *Major issues* dan *Minor issues*.



Tabel 7.8 Hasil *compatibility testing*

No.	Jenis Browser	Versi	Hasil	Status		
				<i>Critical issues</i>	<i>Major issues</i>	<i>Minor issues</i>
1	Internet Explorer (IE)	9	 <p>Gambar 7.6 hasil <i>compatibility testing</i> pada IE versi 9</p>	Lolos	Tidak lolos	Tidak lolos
		10	 <p>Gambar 7.7 hasil <i>compatibility testing</i> pada IE versi 10</p>	Lolos	Tidak lolos	-



Tabel 7.8 Hasil *compatibility testing*

No.	Jenis Browser	Versi	Hasil	Status		
				<i>Critical issues</i>	<i>Major issues</i>	<i>Minor issues</i>
		11	 <p>Gambar 7.8 hasil <i>compatibility testing</i> pada IE versi 11</p>	Lolos	Tidak lolos	-
2	Edge	16	 <p>Gambar 7.9 hasil <i>compatibility testing</i> pada Edge versi 16</p>	Lolos	Lolos	Lolos



Tabel 7.8 Hasil *compatibility testing*

No.	Jenis Browser	Versi	Hasil	Status		
				<i>Critical issues</i>	<i>Major issues</i>	<i>Minor issues</i>
3	Firefox	60	 <p>Gambar 7.10 hasil <i>compatibility testing</i> pada Firefox versi 60</p>	Tidak lolos	-	Tidak lolos
4	Safari	≤ 10	 <p>Gambar 7.11 hasil <i>compatibility testing</i> pada Safari versi ≤ 10</p>	Lolos	Lolos	Lolos



Tabel 7.8 Hasil *compatibility testing*

No.	Jenis Browser	Versi	Hasil	Status		
				<i>Critical issues</i>	<i>Major issues</i>	<i>Minor issues</i>
		11	 <p>Gambar 7.12 hasil <i>compatibility testing</i> pada Safari versi 11</p>	Lolos	Lolos	Lolos
5	Opera	51	 <p>Gambar 7.13 hasil <i>compatibility testing</i> pada Opera versi 51</p>	Lolos	Lolos	Lolos



Tabel 7.8 Hasil *compatibility testing*

No.	Jenis Browser	Versi	Hasil	Status		
				<i>Critical issues</i>	<i>Major issues</i>	<i>Minor issues</i>
6	Chrome	66	 <p>Gambar 7.14 hasil <i>compatibility testing</i> pada Chrome versi 66</p>	Tidak lolos	-	-
7	iOS	≤ 9	 <p>Gambar 7.15 hasil <i>compatibility testing</i> pada iOS versi ≤ 9</p>	Lolos	Lolos	Lolos



Tabel 7.8 Hasil *compatibility testing*

No.	Jenis Browser	Versi	Hasil	Status		
				<i>Critical issues</i>	<i>Major issues</i>	<i>Minor issues</i>
		10	 <p>Gambar 7.16 hasil <i>compatibility testing</i> pada <i>iOS</i> versi 10</p>	Lolos	Lolos	Lolos
		11	 <p>Gambar 7.17 hasil <i>compatibility testing</i> pada <i>iOS</i> versi 11</p>	Lolos	Lolos	Lolos

Tabel 7.8 Hasil *compatibility testing*

No.	Jenis Browser	Versi	Hasil	Status		
				<i>Critical issues</i>	<i>Major issues</i>	<i>Minor issues</i>
8	Android	≤ 3	 <p>Gambar 7.18 hasil <i>compatibility testing</i> pada Android versi ≤ 3</p>	Lolos	Lolos	Lolos
		4	 <p>Gambar 7.19 hasil <i>compatibility testing</i> pada Android versi 4</p>	Lolos	Lolos	Lolos

Tabel 7.8 Hasil *compatibility testing*

No.	Jenis Browser	Versi	Hasil	Status		
				<i>Critical issues</i>	<i>Major issues</i>	<i>Minor issues</i>
9	BlackBerry	≤ 7.1	 <p>Gambar 7.20 hasil <i>compatibility testing</i> pada BlackBerry versi ≤ 7.1</p>	Lolos	Lolos	Tidak lolos
		10	 <p>Gambar 7.21 hasil <i>compatibility testing</i> pada BlackBerry versi 10</p>	Lolos	Lolos	Lolos

7.6.3 Hasil *User Acceptance Testing* (UAT)

7.6.3.1 Hasil UAT untuk validasi sistem

Tabel 7.3 berikut merupakan hasil UAT validasi fungsionalitas dari aktor Tenaga Kesehatan. Kolom kode pengujian berisi kode penanda kasus uji. Kolom kode *use case* berisi kode dari *use case* yang diuji, ditulis untuk memudahkan peruntutan dengan dokumentasi perancangan yang telah dibuat. Sedangkan Tabel 7.4 merupakan hasil kasus uji UAT validasi fungsionalitas aktor dari pihak Dinkes yaitu Petugas Proses; Kepala Seksi Kefarmasian, Kepala Seksi SDM; UPT Puskesmas Wilayah, Seksi Kesehatan Keluarga dan Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas.



Tabel 7.9 Hasil uji UAT validasi aktor Tenaga Kesehatan

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang Diharapkan	Hasil (Valid/Tidak valid)
U-VAL-01-01	UCT-01	Membuat Akun	Pada halaman pertama sistem, penguji memilih fungsi mendaftarkan diri, kemudian memasukkan data diri seperti nama lengkap, Nomor Induk Kependudukan (NIK), <i>email</i> , nomor <i>handphone</i> , <i>password</i> dan jenis kelompok Tenaga Kesehatan. Setelah mengisi, centang pernyataan kebenaran data. Kemudian pilih tombol Daftar.	Penguji dapat berhasil membuat akun dan masuk ke halaman beranda untuk Tenaga Kesehatan.	Valid
			Pada halaman pertama sistem, penguji memilih fungsi mendaftarkan diri, kemudian memasukkan data diri dengan email yang pernah didaftarkan sebelumnya.	Penguji mendapatkan pesan gagal registrasi karena email telah terdaftar.	Valid
U-VAL-01-02	UCT-02	Login	Pada halaman utama sistem, penguji memasukkan Nomor Induk Kependudukan (NIK) dan <i>password</i> dengan hak akses Tenaga Kesehatan, lalu menekan tombol <i>Login</i>	Penguji dapat masuk ke halaman beranda untuk Tenaga Kesehatan.	Valid
			Pada halaman utama sistem, penguji memasukkan Nomor Induk Kependudukan (NIK) dan <i>password</i> yang tidak sesuai.	Penguji mendapatkan pesan gagal <i>login</i> karena NIK dan <i>password</i> tidak sesuai.	Valid

Tabel 7.9 Hasil uji UAT validasi aktor Tenaga Kesehatan

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang Diharapkan	Hasil (Valid/Tidak valid)
U-VAL-01-03	UCT-03	Merubah Biodata	Pada halaman berisi formulir perubahan biodata, penguji mengisi sesuai data yang ingin dirubah. Setelah itu pilih tombol simpan.	Penguji mendapatkan pesan berhasil merubah biodata.	Valid
			Perubahan yang dilakukan gagal disimpan oleh sistem.	Penguji mendapatkan pesan gagal merubah biodata.	Valid
U-VAL-01-04	UCT-09	Merubah Foto	Pada halaman berisi formulir perubahan biodata, penguji memilih tombol ganti foto. Kemudian tekan tombol untuk memilih <i>file</i> . Jika foto telah dipilih, tekan tombol simpan.	Penguji mendapatkan pesan berhasil merubah foto dan kembali ke halaman ubah biodata.	Valid
U-VAL-01-05	UCT-04	Membuat Berkas Pendukung	Penguji memilih menu Berkas Pendukung. Setelah formulir pembuatan berkas ditampilkan, isikan data yang diperlukan sesuai dengan jenis berkas yang akan dibuat, lalu pilih tombol Cetak.	Penguji mendapatkan berkas pendukung yang telah terisi.	Valid
U-VAL-01-06	UCT-05	Mengajukan Permohonan	Pada halaman berisi formulir pengajuan permohonan, penguji mengisikan data dan berkas persyaratan yang sesuai dengan masing-masing jenis kelompok Tenaga Kesehatan. Setelah mengisi, tekan tombol Ajukan Permohonan.	Penguji mendapatkan pesan berhasil mengirimkan permohonan surat izin kerja/praktik.	Valid

Tabel 7.9 Hasil uji UAT validasi aktor Tenaga Kesehatan

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang Diharapkan	Hasil (Valid/Tidak valid)
U-VAL-01-07	UCT-06	Melihat Alur Penanganan	Penguji memilih tombol Beranda.	Penguji berada pada halaman beranda yang berisi tabel <i>log</i> penanganan permohonan.	Valid
U-VAL-01-08	UCT-07	Mencetak Tanda Bukti	Pada tabel berisi <i>log</i> status permohonan yang diajukan, penguji memilih tombol Cetak Tanda Bukti jika permohonan telah disetujui petugas Dinkes.	Penguji mendapatkan tanda bukti terima berkas.	Valid
U-VAL-01-09	UCT-08	<i>Logout</i>	Penguji memilih tombol <i>Logout</i> .	Penguji kembali ke halaman <i>login</i> .	Valid

Tabel 7.10 Hasil uji UAT validasi fungsional aktor Petugas Proses; Kepala Seksi Kefarmasian, Kepala Seksi SDM; UPT Puskesmas Wilayah, Seksi Kesehatan Keluarga dan Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas

Kode Pengujian	Kode Use Case	Fungsional Sistem	Kasus Uji	Yang Diharapkan	Hasil (Valid/Tidak valid)
U-VAL-02-01	UCP-01	Login	Pada halaman utama sistem, penguji memasukkan <i>username</i> dan <i>password</i> , lalu menekan tombol <i>Login</i> .	Penguji dapat berhasil membuat akun dan masuk ke halaman beranda untuk petugas Dinkes.	Valid
			Pada halaman utama sistem, penguji memasukkan <i>username</i> dan <i>password</i> yang tidak sesuai, lalu menekan tombol <i>Login</i> .	Penguji mendapatkan pesan gagal <i>login</i> karena <i>username</i> dan <i>password</i> tidak sesuai.	Valid
U-VAL-02-02	UCP-02	Menyetujui Pengajuan Permohonan	Pada halaman Detil Permohonan, penguji memilih menerima atau tidak, mengisi catatan tambahan, memilih lengkap/tidaknya berkas, mengisi kekurangan berkas bila ada, kemudian pilih tombol Kirim.	Penguji mendapatkan pesan berhasil mengirimkan persetujuan pengurusan permohonan.	Valid
U-VAL-02-03	UCP-03	Menyetujui Penanganan Lebih Lanjut	Pada halaman Detil Permohonan, penguji mengisi catatan tambahan, kemudian pilih tombol Kirim.	Penguji mendapatkan pesan berhasil mengirimkan persetujuan untuk penanganan lebih lanjut.	Valid
U-VAL-02-04	UCP-04	Membuat Jadwal Kunjungan	Pada halaman Detil Permohonan, penguji memasukkan tanggal dan jam kunjungan pada formulir buat jadwal, kemudian pilih tombol Simpan.	Penguji mendapatkan pesan berhasil membuat jadwal kunjungan visitasi.	Valid

Tabel 7.10 Hasil uji UAT validasi fungsional aktor Petugas Proses; Kepala Seksi Kefarmasian, Kepala Seksi SDM; UPT Puskesmas Wilayah, Seksi Kesehatan Keluarga dan Gizi; Kepala Bidang SDM, Kefarmasian dan Alat Kesehatan; Kepala Dinas

Kode Pengujian	Kode <i>Use Case</i>	Fungsional Sistem	Kasus Uji	Yang Diharapkan	Hasil (Valid/Tidak valid)
U-VAL-02-05	UCP-05	Melihat Daftar Jadwal Kunjungan	Penguji memilih tombol Jadwal Kunjungan.	Penguji berhasil melihat daftar jadwal kunjungan yang telah dibuat.	Valid
U-VAL-02-06	UCP-06	Membuat Berita Acara	Pada halaman Detil Permohonan, penguji mengisi setiap poin yang ada pada formulir, kemudian pilih tombol Kirim.	Penguji mendapatkan pesan berhasil membuat berita acara.	Valid
U-VAL-02-07	UCP-07	Menugaskan Pembuatan Surat Izin	Pada halaman Detil Permohonan, penguji mengisi catatan tambahan, kemudian pilih tombol Kirim.	Penguji mendapatkan pesan berhasil mengirimkan penugasan pencetakan surat izin.	Valid
U-VAL-02-08	UCP-08	Mencetak Surat Izin	Pada halaman detil Permohonan, penguji memilih tombol Cetak Surat.	Penguji mendapatkan surat izin kerja/praktik yang telah terisi oleh sistem.	Valid
U-VAL-02-09	UCP-09	Pengujian <i>Logout</i>	Penguji memilih tombol <i>Logout</i> .	Penguji kembali ke halaman <i>login</i> petugas Dinkes.	Valid

Dari hasil pengujian validasi yang ditunjukkan pada Tabel 7.3 dan 7.4 diketahui bahwa semua fungsionalitas baik milik aktor Tenaga Kesehatan dan pihak Dinkes telah berjalan sesuai dengan perancangan yang telah dibuat dan sistem dapat berjalan sesuai dengan yang diharapkan. Setelah memastikan bahwa sistem berjalan dengan harapan, maka selanjutnya dilakukan pengujian penerimaan pengguna yang diwakilkan oleh tiga pegawai CV. Sarana Utama Solusindo.

7.6.3.2 Hasil UAT dampak setelah adanya sistem

Berikut pada Tabel 7.5 ditunjukkan hasil kuesioner penilaian yang didapatkan untuk pengujian UAT pengajuan permohonan. Sedangkan Tabel 7.6 merupakan hasil kuesioner penilaian yang didapatkan untuk UAT penanganan permohonan. Hasil pengujian penerimaan dari sisi Tenaga Kesehatan didapatkan dari 4 responden yang terdiri dari Apoteker, Dokter, Perawat dan Tenaga Teknis Kefarmasian. Kemudian untuk penerimaan dari pihak Dinkes didapatkan dari pengujian yang dilakukan oleh 3 orang perwakilan CV. Sarana Utama Solusindo. Nilai-nilai yang ada merupakan hasil penjumlahan jawaban yang didapatkan untuk setiap pertanyaan.

Tabel 7.11 Hasil kasus uji UAT pengajuan permohonan

No	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1.	Apakah SINKES dapat diakses dengan cepat?	U-UAT-01-01	0	0	0	2	2
2.	Apakah SINKES mudah digunakan?	U-UAT-01-02	0	0	1	2	1
3.	Apakah tampilan SINKES ketika digunakan?	U-UAT-01-03	0	0	2	1	1
4.	Apakah SINKES mempermudah penyampaian informasi penanganan antar petugas dan Kepala Dinas?	U-UAT-01-04	0	0	1	2	1
5.	Apakah dengan adanya SINKES membantu dalam proses pengajuan permohonan surat izin kerja atau praktik?	U-UAT-01-05	0	0	0	3	1

Tabel 7.11 Hasil kasus uji UAT pengajuan permohonan

No	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
6.	Apakah dengan adanya SINKES membantu dalam mengetahui penanganan permohonan yang telah diajukan?	U-UA6-01-06	0	0	0	1	3
7.	Apakah login user dibutuhkan untuk membantu keamanan informasi pemohon dalam SINKES?	U-UAT-01-07	0	0	2	1	1
Total			0	0	6	12	10

Hasil penilaian pada Tabel 7.5 selanjutnya dilakukan perhitungan menggunakan skala *Likert* dengan bobot nilai masing-masing 5 untuk jawaban Sangat Setuju (A), 4 untuk jawaban Setuju (B), 3 untuk jawaban Netral (C), 2 untuk jawaban Tidak Setuju (D) dan 1 untuk jawaban Sangat Tidak Setuju (E). Tabel 7.6 berikut menunjukkan perhitungan dari jawaban yang didapat untuk UAT pengajuan permohonan.

Tabel 7.12 Tabel hasil kalkulasi skala *Likert* UAT pengajuan permohonan

Jawaban	Bobot nilai	Jumlah jawaban (n)	Hasil
Sangat Setuju (A)	5	5	25
Setuju (B)	4	14	56
Netral (C)	3	2	6
Tidak Setuju (D)	2	0	0
Sangat Tidak Setuju (E)	1	0	0
Total			87

Selanjutnya pada perhitungan skala *likert* yang diperoleh dari hasil kuisioner *user acceptance testing* pengajuan permohonan dengan menghitung nilai Y.

$$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$$

$$Y = 5 \times 4 \times 7 = 140$$

Kemudian pada akhir perhitungan skala *likert* yang diperoleh dari hasil kuisioner *user acceptance testing* pengajuan permohonan dengan menggunakan rumus index untuk menghitung nilai persentase

$$\text{Rumus Index} = (\text{Total Nilai} \div Y) \times 100\%$$

$$\text{Rumus Index} = (116 \div 140) \times 100\% = 82.9\% \text{ (Sangat Setuju)}$$

Tabel 7.13 Hasil kasus uji UAT penanganan permohonan

No	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
1.	Apakah SINKES dapat diakses dengan cepat?	U-UAT-02-01	0	0	0	3	0
2.	Apakah SINKES mudah digunakan?	U-UAT-02-02	0	0	0	2	1
3.	Apakah tampilan SINKES nyaman ketika digunakan?	U-UAT-02-03	0	0	0	3	0
4.	Apakah SINKES mempermudah penyampaian informasi penanganan antar petugas dan Kepala Dinas?	U-UAT-02-04	0	0	0	1	2
5.	Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan membantu dalam pencatatan permohonan masuk?	U-UAT-02-05	0	0	0	0	3
6.	Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan membantu pemeriksaan berkas pemohon?	U-UAT-02-06	0	0	0	2	1
7.	Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan membantu dalam memberikan	U-UAT-02-07	0	0	0	3	0

Tabel 7.13 Hasil kasus uji UAT penanganan permohonan

No	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
	pemberitahuan kepada Tenaga Kesehatan?						
8.	Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan membantu dalam melakukan tindakan penanganan?	U-UAT-02-08	0	0	0	2	1
9.	Apakah dengan adanya sistem dapat membantu dalam membuat jadwal kunjungan survei?	U-UAT-02-09	0	0	0	2	1
10.	Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan dapat memudahkan melihat jadwal kunjungan survei dan Tenaga Kesehatan yang bersangkutan?	U-UAT-02-10	0	0	0	2	1
11.	Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan dapat membantu pembuatan berita acara kunjungan?	U-UAT-02-11	0	0	0	2	1
12.	Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan dapat membantu penugasan pembuatan surat izin kerja/praktik?	U-UAT-02-12	0	0	0	2	1
13.	Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan dapat membantu mengetahui	U-UAT-02-13	0	0	0	2	1

Tabel 7.13 Hasil kasus uji UAT penanganan permohonan

No	Pertanyaan	Kode Task	Jawaban				
			Sangat Tidak Setuju	Tidak Setuju	Netral	Setuju	Sangat Setuju
	penanganan yang telah diberikan pada masing-masing tahap?						
14.	Apakah dengan adanya Sistem Informasi Izin Kerja dan Praktik Tenaga Kesehatan membantu dalam pengarsipan dokumen permohonan	U-UAT-02-14	0	0	0	1	2
15.	Apakah login user dibutuhkan untuk membantu keamanan informasi petugas dalam SINKES?	U-UAT-02-15	0	0	0	1	2
Total			0	0	0	28	17

Hasil penilaian pada Tabel 7.7 selanjutnya dilakukan perhitungan menggunakan skala *Likert* dengan bobot nilai masing-masing 5 untuk jawaban Sangat Setuju (A), 4 untuk jawaban Setuju (B), 3 untuk jawaban Netral (C), 2 untuk jawaban Tidak Setuju (D) dan 1 untuk jawaban Sangat Tidak Setuju (E). Tabel 7.8 berikut menunjukkan perhitungan dari jawaban yang didapat untuk UAT pengajuan permohonan.

Tabel 7.14 Tabel hasil kalkulasi skala *Likert*

Jawaban	Bobot nilai	Jumlah responden (n)	Hasil
Sangat Setuju (A)	5	17	85
Setuju (B)	4	28	112
Netral (C)	3	0	0
Tidak Setuju (D)	2	0	0
Sangat Tidak Setuju (E)	1	0	0
Total			197

Selanjutnya pada perhitungan skala *likert* yang diperoleh dari hasil kuisisioner *user acceptance testing* penanganan permohonan dengan menghitung nilai Y.

$$Y = \text{Jumlah Skala} \times \text{Jumlah Responden} \times \text{Jumlah Pertanyaan}$$

$$Y = 5 \times 3 \times 15 = 255$$

Kemudian pada akhir perhitungan skala *likert* yang diperoleh dari hasil kuisisioner *user acceptance testing* penanganan permohonan dengan menggunakan rumus index untuk menghitung nilai persentase

$$\text{Rumus Index} = (\text{Total Nilai} \div Y) \times 100\%$$

$$\text{Rumus Index} = (197 \div 255) \times 100\% = 77.3\% \text{ (Setuju)}$$

Pada analisis hasil perhitungan kriteria *user acceptance testing* penanganan permohonan yang telah dihitung dengan menggunakan *skala likert* menghasilkan nilai 82.9% yang berarti dalam interpretasi nilai berarti Tenaga Kesehatan menyatakan sangat setuju dengan SINKES yang sudah dibangun, sedangkan untuk *user acceptance testing* penanganan permohonan nilai 77.3% yang berarti dalam interpretasi nilai dari pihak CV. Sarana Utama Solusindo menyatakan setuju dengan SINKES yang sudah dibangun.

Hasil penilaian yang didapatkan untuk kriteria *performance* menunjukkan bahwa penguji setuju jika sistem dikatakan dapat diakses dengan cepat. Selanjutnya untuk kriteria *usability* pada UAT pengajuan dan penanganan permohonan diketahui bahwa sistem dapat dianggap bagus dan mudah dalam penggunaannya. Kemudian pada kriteria *functional correctness and completeness* menurut penguji sudah diterima dengan baik karena telah sesuai dengan kebutuhan pengajuan maupun penanganan permohonan surat izin yang masuk. Pada kriteria *confidentially and availability* dianggap sangat perlu oleh penguji agar data pribadi milik Tenaga Kesehatan yang melakukan permohonan lebih aman dan akses untuk tiap pengguna dibatasi sesuai kebutuhan.

BAB 8 KESIMPULAN DAN SARAN

8.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Pada tahap analisis kebutuhan, terdapat beberapa aktivitas yang dilakukan. Pertama-tama, dilakukan analisis terhadap 6 jenis proses bisnis terkait pengajuan permohonan sampai penanganan yang sedang berjalan di organisasi (*as-is*). Selanjutnya dilakukan analisis permasalahan, identifikasi proses bisnis usulan (*to-be*) yang diharapkan dapat menyelesaikan permasalahan yang ada, analisis perbedaan antara proses bisnis *as-is* dan *to-be*. Dari identifikasi dengan sudut pandang bisnis (*business perspective*), selanjutnya dilakukan analisis persyaratan dengan hasil berupa dokumentasi yang memuat informasi mengenai hasil identifikasi tipe pemangku kepentingan serta daftar kebutuhan pemangku kepentingan dan pengguna. Kemudian pada identifikasi dari sudut pandang pengguna (*user perspective*), dilakukan analisis fitur dan persyaratan sistem. Dari proses tersebut dihasilkan 9 aktor, 6 fitur, 18 kebutuhan fungsional dan 2 kebutuhan non fungsional. Kebutuhan fungsional yang telah diidentifikasi selanjutnya dimodelkan menggunakan diagram *use case* yang dibedakan menjadi 2 diagram berdasarkan internal Dinkes dan Tenaga Kesehatan untuk memudahkan pendokumentasian. *Use case* yang telah digambarkan selanjutnya dijelaskan dengan rinci menggunakan *use case scenario* dan urutan aktivitas yang dilakukan oleh pengguna divisualisasikan kedalam *activity diagram*.
2. Hasil yang didapatkan pada tahap perancangan Sistem Informasi Pengajuan Izin Kerja Dan Praktik Tenaga Kesehatan (SINKES) yang telah dilakukan berupa pemodelan interaksi antar obyek yang terlibat dengan menggunakan *sequence diagram*, pemodelan obyek yang digambarkan menggunakan ke dalam *class diagram*, *Conceptual Data Model (CDM)*, *Physical Data Model (PDM)* yang menggambarkan 17 tabel sebagai media penyimpanan data dari sistem yang akan dibangun, perancangan antarmuka pengguna, serta algoritme pemrograman yang akan diterjemahkan ke dalam bahasa pemrograman di tahap implementasi.
3. Perancangan sistem yang telah didokumentasikan pada tahap sebelumnya telah diimplementasikan kedalam bahasa pemrograman dengan kerangka kerja *CodeIgniter* hingga tiap fungsinya dapat membantu proses pengajuan permohonan dan penanganan dari permohonan yang telah diterima. Adapun fitur yang telah diimplementasikan adalah fitur dengan kode FEAT01 (Sistem bisa digunakan untuk mengajukan permohonan surat izin kerja/praktik), FEAT02 (Sistem bisa digunakan untuk menangani permohonan yang telah

diajukan oleh Tenaga Kesehatan), FEAT04 (Sistem dapat memberikan otoritas pada pengguna sesuai dengan peran dan tanggungjawab yang dimiliki), FEAT05 (Sistem dapat ditampilkan dengan baik di berbagai ukuran layar perangkat) dan FEAT06 (Sistem dapat berjalan dengan baik pada berbagai jenis *browser*).

4. Terdapat tiga jenis metode pengujian yang telah dilakukan, yaitu *responsiveness testing*, *compatibility testing* dan *User Acceptance Testing* (UAT). Pertama, *responsiveness testing* dilakukan pada tiga ukuran resolusi layar yang berbeda yaitu layar *desktop* (1440*900), tablet (768*1024) dan telepon genggam (414*736). Pengujian pada ketiga ukuran layar tersebut memperlihatkan bahwa tampilan antarmuka sistem informasi yang telah dikembangkan dapat beradaptasi dengan baik di masing-masing ukuran layar. Selanjutnya, dilakukan *compatibility testing* dengan menggunakan alat bantu perangkat lunak *Sortsite* versi *desktop* yang menunjukkan bahwa terdapat *Critical Issues* pada jenis *browser Firefox* 60 dan *Chrome* 66, kemudian *Major Issues* yang terjadi pada jenis *browser IE (Internet Explorer)* versi 9, 10 dan 11 dan *Minor Issues* yang terjadi pada *browser IE (Internet Explorer)* versi 9, *Firefox* 60 dan *Blackberry* ≤ 7. Kemudian untuk hasil *User Acceptance Testing* (UAT) yang dilakukan oleh 4 Tenaga Kesehatan dan perwakilan 3 pegawai CV. Sarana Utama Solusindo, diketahui bahwa sistem yang terbagi menjadi dua sisi yaitu Tenaga Kesehatan (pengajuan permohonan) dan pihak Dinkes (penanganan permohonan) menunjukkan hasil yang berbeda. Hasil tersebut menunjukkan bahwa sistem dari sisi Tenaga Kesehatan (pengajuan permohonan) memiliki nilai akhir penerimaan pengguna sebesar 82.9% (Sangat setuju), sedangkan dari sisi pihak Dinkes (penanganan permohonan) sebesar 77.3% (Setuju).

Dari kesimpulan yang telah dijabarkan, maka secara umum Sistem Informasi Pengajuan Izin Kerja Dan Praktik Tenaga Kesehatan (SINKES) dapat mendemonstrasikan fungsi-fungsi yang telah dibangun untuk mendukung proses pengajuan permohonan Surat Izin Kerja/Praktik dan penanganan terhadap permohonan yang diajukan.

8.2 Saran

Saran yang diberikan penulis guna penelitian Sistem Informasi Pengajuan Izin Kerja Dan Praktik Tenaga Kesehatan (SINKES) lebih lanjut yang dapat dilakukan adalah sebagai berikut.

1. Dapat dilakukan penambahan fungsi *sms* dan *email gateway* sebagai sarana tambahan dalam mengirimkan pemberitahuan kemajuan penanganan permohonan kepada Tenaga Kesehatan yang bersangkutan, agar pemberitahuan dapat segera didapatkan tanpa Tenaga Kesehatan perlu untuk melakukan *login* sistem terlebih dahulu.

2. Dapat ditambahkan fungsional *dashboard* bagi admin sebagai bentuk visualisasi pengolahan data yang didapatkan dari awal proses pengajuan permohonan hingga surat izin diterima oleh Tenaga Kesehatan.
3. Dapat ditambahkan fungsi yang memungkinkan penjadwalan kunjungan secara otomatis ketika Kepala Bidang menyetujui pembuatan berkas. Penjadwalan otomatis dapat dilakukan berdasarkan jarak antar tempat praktik dan tanggal pengajuan dari Tenaga Kesehatan.
4. Dapat ditambahkan fungsi untuk *report* bulanan sebagai bentuk tanggungjawab petugas kepada Kepala Dinas.
5. Dapat dilakukan evaluasi antarmuka pengguna untuk mengetahui tingkat *usability* bagi pengguna, sebagai dasar melakukan perbaikan di pengembangan selanjutnya.



DAFTAR PUSTAKA

- Balaji, S., 2012. Waterfall VS V-Model VS Agile : A Comparative Study on SDLC. *International Journal of Information Technology and Business Management*, II(1).
- Bertolino, A., 2007. Software Testing Research: Achievements, Challenges, Dreams. *Future of Software Engineering*.
- Bertram, D., 2016. *Likert Scale*. s.l.:s.n.
- Bittner, K. & Spence, I., 2002. *Use Case Modeling*. Boston: Addison Wesley.
- Booch, G. et al., 2007. *Object-Oriented Analysis and Design with Applications*. 3rd ed. Boston: Addison-Wesley.
- Eka, W., Bukhori, S. & Ismoyo, D., 2013. *Perbandingan V-Model Tradisional dan Advance V-Model*, Samarinda: Seminar Nasional Ilmu Komputer.
- Hambling, B. & Goethem, P. v., 2013. *User Acceptance Testing: A Step-by-step Guide*. Swindon: BCS Learning & Development Limited.
- Hustinawati, Himawan, A. K. & Latifah, 2014. Performance Analysis Framework Codeigniter and CakePHP in Website Creation. *International Journal of Computer Applications (0975 – 8887)*, Volume 94, p. 7.
- Ishami, F., Rokhmawati, R. I. & Saputra, M. C., 2017. Pengembangan Sistem Self-Service Reservation pada Everyday Smart Hotel Malang Menggunakan Customer Relationship Management (CRM) Operasional Modul Sales Force Automation dan Service Automation. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, II(2), pp. 632-640.
- Kurniawan, T. A., 2018. Pemodelan Use Case (UML) : Evaluasi Terhadap Beberapa Kesalahan dalam Praktik. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 5(1), pp. 77-86.
- Mathur, S. & Malik, S., 2010. Advancements in the V-Model. *International Journal of Computer Applications*, I(12).
- Naik, K. & Tripathy, P., 2008. *Software Testing and Quality Assurance Theory and Practice*. Hoboken: John Wiley & Sons, Inc..
- Nurjanah, A. S. & Kurniadi, D., 2017. Sistem Informasi Pengelolaan Izin Praktek Kerja Lapangan untuk Sekolah Menengah Kejuruan Secara Online di STT Garut. *Jurnal STT Garut*, 14(2).
- Object Management Group, 2011. *Business Process Model and Notation (BPMN)*. [Online]
Available at: <http://www.omg.org/spec/BPMN/2.0/PDF>
[Accessed 17 Agustus 2017].
- Petersen, K., Wohlin, C. & Baca, D., 2009. The Waterfall Model in Large-Scale Development. *Springer-Verlag Berlin Heidelberg*, pp. 386-400.

PowerMapper, 2017. *SortSite - Browser Compatibility Tests*. [Online]
Available at: <https://www.powermapper.com/products/sortsite/checks/browser-compatibility/>

[Accessed 30 Mei 2018].

Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach*. 7th ed.
New York: The McGraw-Hill Companies, Inc.

Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach*. 7th ed.
New York: McGraw-Hill.

Sawant, A. A., Bari, P. H. & Chawan, P. M., 2012. Software Testing Techniques and Strategies. *International Journal of Engineering Research and Applications (IJERA)*, 2(3).

Sharma, P. & Singh, D., 2015. Comparative Study of Various SDLC Models on Different Parameters. *International Journal of Engineering Research*, IV(4), pp. 188-191.

Sommerville, I., 2009. *Software Engineering*. 9th ed. Boston: Addison-Wesley.

Sparx Systems, 2017. *Sparx Systems*. [Online]
Available at: <http://www.sparxsystems.com.au/resources/user-guides/model-domains/database-models.pdf>

[Accessed 8 Februari 2018].

Sugiyono, 2016. *Metode Penelitian Kombinasi (Mixed Methods)*. 8 ed. Bandung: Alfabeta.

Sukamto, R. A. & Shalahuddin, M., 2016. *Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Obyek)*. Bandung: Informatika.

Swati, 2018. *Software Testing Help*. [Online]
Available at: <https://www.softwaretestinghelp.com/responsive-web-design-testing-guide>

[Accessed 30 May 2018].